

1-1-2008

Image Processing Techniques For Improved Sun Sensor Performance

Christopher. Li
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Li, Christopher., "Image Processing Techniques For Improved Sun Sensor Performance" (2008). *Theses and dissertations*. Paper 1149.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

TK
8315
LS
2008

Image Processing Techniques for Improved Sun Sensor Performance

by

Christopher Li

Bachelor of Engineering, Aerospace Engineering,
Ryerson University, 2006

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the program of
Mechanical Engineering

**PROPERTY OF
RYERSON UNIVERSITY LIBRARY**

Toronto, Ontario, Canada, 2008

©Christopher Li 2008

Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Image Processing Techniques for Improved Sun Sensor Performance

Christopher Li

Master of Applied Science
Program of Mechanical Engineering, Ryerson University, 2008

Abstract

The behaviour of digital sun-sensors and associated super-resolution algorithms was explored. Using calibration data, a method was proposed to model the peak width of peaks across the image array. Using this with the non-linear least square algorithm gave improved performance across the field-of-view. A test was proposed that would measure precision for small sensor motions. Also, a method of accounting for local bias error was given. The small motion test defined limits at which the sensor detects motion, and the precision test gave metrics to measure how well the sensor renders motion. Finally, an extended kalman filter was developed that used sun-vector measurements, in addition to a new relative measurement. This was tested using a well-defined sensor as well as a generic sensor for which few error data were known. Results indicate that relative measurements only improve performance if random noise is low.

Acknowledgements

I would first like to thank my thesis supervisor, Dr. John Enright, for giving me the opportunity to pursue this research. His guidance and encouragement have given me the means to investigate these topics.

I would also like to thank Mr. Doug Sinclair from Sinclair Interplanetary for providing the SS-256 and SS-411 sun sensors; also, for the technical assistance he has given our research group throughout the last two years.

A thank you also goes out to Dr. Don McTavish for his assistance in helping tackle the issues in building a dynamics simulator.

I would also like to thank Albert Yam for his help throughout the past two years. He helped me understand signal processing.

I would like to thank my father, Tony Li, and my mother, Lynda Meandro, for their unwavering support of my academic career, and their words of encouragement whenever I felt overwhelmed. Finally, I would like to thank Sinah Lee. Thank you for your love and support. You've stood by me throughout my entire university career, and without you I don't believe I could ever have dreamed of getting this far.

Contents

Declaration	iii
Abstract	v
Acknowledgements	vii
Table of Contents	xii
List of Figures	xiv
List of Tables	xv
Nomenclature	xvi
1 Introduction	1
1.1 Attitude Estimation Methods	2
1.1.1 Gyros and Accelerometers	2
1.1.2 Horizon Sensors	3
1.1.3 Magnetometers	3
1.1.4 Star Trackers	3
1.1.5 Sun Sensors	4
1.1.6 Comparison of Systems	4
1.2 Sun Sensor Basics	5
1.2.1 Analog Sun Sensors	5

1.2.2	Digital Sun Sensors	5
1.2.3	Digital Sun Sensor Improvement	6
1.3	Work Objectives	7
1.3.1	Thesis Outline	8
2	Background	9
2.1	Sun Sensor Optics	9
2.2	Identifying the Image Characteristics	14
2.3	Super-Resolution	16
2.3.1	First-Order Centroiding	17
2.3.2	Non-Linear Least Squares	17
2.3.3	Linear-Phase	20
2.4	Algorithm Testing and Sensor Calibration	22
2.4.1	Calibration Approach	22
2.4.2	The Sun Sensor Test Setup	23
2.4.3	Calibration Image Sets	23
2.4.4	Calibration Procedure	26
3	Peak Width Correction	29
3.1	Peak Shape Theory and Data Collection	29
3.1.1	Theory Development	29
3.1.2	Width Data Collection	30
3.2	Relating Peak Width to Array Position	32
3.2.1	Fit Equation Development	32
3.2.2	Curve Fit Results	34
3.3	Calibration Revision	35
3.3.1	Index of Refraction	35
3.3.2	Longitudinal Gap	35
3.3.3	Transverse Gap	36
3.4	Effects on Algorithm Performance	36
3.5	Summary	39

4	Precision Testing	41
4.1	Motion Limit	42
4.1.1	Motion Limit Theory	43
4.1.2	Motion Limit Results	45
4.2	Calibration Offsets and the Best Rotation Quaternion	45
4.2.1	Quaternion Mapping Theory	46
4.2.2	Applying the Quaternion	48
4.3	Precision	50
4.3.1	Definitions in the Precision Test	51
4.3.2	Precision Test Results	52
4.4	Summary	55
5	Filtering for Rate Sensing	57
5.1	Filter Formulation	58
5.2	Defining the System	60
5.2.1	Derivation of the Expected Observation	62
5.2.2	Deriving F and H	63
5.2.3	Sensor Noise Modelling	65
5.2.4	Planned Measurement Simulation	66
5.2.5	Initialization Values	66
5.3	Exact Solution	67
5.4	Generic Sensor Simulation	68
5.4.1	Sun-Vector Observation Only	68
5.4.2	Relative Observation	69
5.5	SS-411 Simulation	71
5.5.1	Sun-Vector Observation Only	71
5.5.2	Relative Observation	72
5.6	EKF Remarks	74
6	Conclusions	75
6.1	Summary	75

6.1.1	Calibration Work	76
6.1.2	Width Correction	76
6.1.3	Precision Testing	77
6.1.4	Extended Kalman Filter	78
6.2	Future Work	79
6.2.1	Field-of-View Precision	79
6.2.2	EKF/Precision Model Unification	80
6.3	Closing Remarks	80
	Bibliography	81

List of Figures

2.1	Geometric Model of Sensor Showing Aperature and Image Planes.[Enright, 2008]	11
2.2	Geometric Model of the Image Plane.	12
2.3	Image of the Central Array Gap.[Enright, 2008]	12
2.4	Four-Peak versus Three-Peak Image.	15
2.5	Experimental Setup.	24
3.1	Image Peak with Curve Fit.	31
3.2	Fitted Peak Width by Array Position.	32
3.3	Peak Width by Array Position with Curve Fit.	34
3.4	Definition of Longitudinal and Transverse Gaps.	36
3.5	FOV Error Plot with No Width Correction (in Radians).	37
3.6	FOV Error Plot with Width Correction (in Radians).	38
4.1	Confidence that Motion has Occurred.	46
4.2	Difference Between Estimated and True Sun Vectors Before Bias Reduction.	47
4.3	Difference Between Estimated and True Sun Vectors After Bias Removal.	50
4.4	Typical FOV Error Plot (in Radians).	51
4.5	Local Region RMS Error.	53
4.6	Far Field RMS Error.	53
4.7	Local Region Normalized RMS Error.	54

4.8	Normalized RMS Error to 1°Offset.	55
5.1	Error Vector Magnitude in \hat{s} for Generic Sensor, No Relative Observation.	69
5.2	X-axis relative observation Comparison.	70
5.3	Error Vector Magnitude in \hat{s} for Generic Sensor with Relative Observation.	71
5.4	Error Vector Magnitude in \hat{s} for SS-411, No Relative Obser- vation.	72
5.5	X-axis relative observation Comparison.	73
5.6	Error Vector Magnitude in \hat{s} for SS-411 with Relative Obser- vation.	74

List of Tables

1.1	ADCS Sensor Comparison(Adapted from [Wertz, 1999])	4
2.1	Physical Sensor Paramaterization	27
3.1	Fourth-Order Curve Fit Parameters	34
3.2	Calibration Summary	39

Nomenclature

Symbol	Description
\hat{s}	Sun Vector
ϕ	Array-Mask Offset
D	Slit Width of x-axis Slits
γ	Pixel Length
m	Array Position (in pixels)
ρ_x	Pixel-0 x-Location
ψ	Image-Array Angle
n_{glass}	Index of Refraction of Mask
h	Array-Mask Distance
Ω	Subtended Area
ζ	Annular Offset
$I[n]$	Image on Array
\vec{r}_1	Platform Center of Rotation to Sun
\vec{r}_2	Platform Center of Rotation to Sensor
	Origin
τ	Array Location
σ	Standard Deviation
ε	Error
G	Longitudinal Gap Length
G_T	Transverse Gap Length
θ	Angle Between Two Vectors

μ	Mean Value
ν	Degrees of Freedom
α	Confidence Level
\vec{q}	Quaternion
$f()$	System Equations of Motion
$Z()$	Observation
$h()$	Expected Observation
$K()$	Filter Gain Matrix
$P()$	Covariance Matrix
$X()$	State Matrix
w_x	Angular Velocity about x-axis
I_{xx}	x-axis Moment of Inertia

Chapter 1

Introduction

Attitude determination and control systems (ADCS) is a critical part of space vehicle design. A number of disturbance torques act on any space vehicle, and many of these are unpredictable and change with time. The responsibility of the ADCS is to observe the effects of these disturbances and align the space vehicle as desired, rejecting the external effects that attempt to change the orientation away from this desired state.

Attitude determination is made using various different types of sensors. Although many different phenomena can be observed to determine the attitude of the spacecraft, sensors and associated information processing will return one of two things. Some sensors yield an inertial measurement, namely, the current alignment of the space vehicle with respect to some external reference. Another class of sensors returns relative measurements, such as the angular momentum of the vehicle or the rate of change of the orientation. Some sensors, such as star trackers, can operate in both modes, and often multiple types of sensors are used for situational redundancy.

In this section, a general comparison of different attitude determination sensors will be performed. These different sensor classes can be used in conjunction with sun sensors in different determination systems to deliver better overall performance. Once the role of sun sensors with respect to

1. INTRODUCTION

other sensor types is understood, a more in-depth description of how sun sensors work will be given. Finally, the objectives of this thesis regarding sun sensor improvement will be defined.

1.1 Attitude Estimation Methods

A number of different types of sensors are available that yield a measure of the space vehicle's orientation. There are a number of ways in which these sensors differ. Selection of the appropriate sensor requires an analysis of the accuracy requirements of the mission, the cost, mass, and power budgets allotted to the system, and the conditions in which the sensors are expected to operate.

In many cases, multiple attitude sensing devices can be used so that attitude observability is maintained when certain attitude sensors aren't usable. For example, a sun-sensor cannot be used when the sun is outside the field-of-view, or when the sun is eclipsed – usually by the earth. In this case, another sensing method must be available to receive attitude information.

1.1.1 Gyros and Accelerometers

Gyros and accelerometers measure the speed of rotation about a particular axis. When an initial reference point is given, or is found using other sensor types, accelerometer data can be used to update the estimate through time. Because no absolute position measurements can be taken, the attitude estimates will eventually drift away from the true value in absence of other measurement types. Depending on the type of sensor used, this drift can vary from 0.003° up to 1° every hour[Wertz, 1999].

1.1.2 Horizon Sensors

Horizon sensors are infrared sensors that detect the heat given off by the earth's atmosphere. This allows the spacecraft to know its orientation relative to the earth-disc. Accuracy for these systems runs from 0.1° to 0.25° . Generally, the use of horizon sensors requires unobstructed view of the earth through the field-of-view cone.

1.1.3 Magnetometers

Magnetometers are very simple and lightweight devices that yield crude attitude estimates by detecting the surrounding magnetic field. When this is compared with the known earth's magnetic field, some attitude information can be found. Magnetometers are often used in conjunction with higher-precision sensors, such as star or sun sensors, for increased accuracy.

1.1.4 Star Trackers

Star sensors are high-precision devices that work by looking at star positions within the field-of-view. They can be used for differential *tracking*, where the relative movement of the bright spots in the field of view can be used to derive a rotational speed. If uploaded with a star catalogue and more sophisticated software, a star sensor can provide a high-accuracy measurement of absolute orientations by comparing the picture taken with known star patterns.

Although they are very high-precision devices, star sensors suffer a number of drawbacks. They are generally heavier than many other sensor types, use more power, and are of higher cost. They also can be blinded by a number of things, including the sun, the earth, and other planets - since they often cause star pattern identification failures. Nonetheless, star sensors are often used when high-precision applications are considered.

1. INTRODUCTION

1.1.5 Sun Sensors

Sun sensors are lightweight, low-cost, low-power devices that offer fairly good resolution. Typically, they return the instantaneous sun-vector direction relative to the camera center. This allows us to determine two components of the spacecraft orientation. Determination of the third requires another detector.

Highly accurate observations require that the sun sensor have an unobstructed view of the sun. Since they are generally low mass, low cost devices, multiple sensors are often mounted around the spacecraft so that observation data can be returned regardless of the spacecraft orientation. However, prolonged eclipse periods (usually due to the earth) require that some other form of attitude recognition be present.

1.1.6 Comparison of Systems

A comparison of the different attitude determination systems is shown in Table(1.1). This is designed to give a general overview only – The actual accuracy depends on the design of the sensor.

Table 1.1: ADCS Sensor Comparison(Adapted from [Wertz, 1999])

Sensor	Accuracy	Cost
Gyros & Accelerometers	High	High
Sun Sensors	Med.	Low
Star Sensors	High	High
Horizon Sensors	Med.	Med.
Magnetometers	Low	Low

1.2 Sun Sensor Basics

Since the sun itself is easy to distinguish from other stellar objects, it is a useful tool for determining orientation. Some sun sensors take single angular measurements, others are capable of two. Because the rotations about the observed sun-vector are not observable, some other method is required to achieve full attitude observability[Godard, 2006]. Sun sensors themselves are further categorized as *analog* or *digital*.

1.2.1 Analog Sun Sensors

Analog sun sensors work by comparing the relative signal obtained by relatively few cells on a two-dimensional matrix. The amount of light intensity received by the cells will be proportional to the off-boresight angle of the sun. Digital sensors use a larger number of photosensitive cells in conjunction with a patterned mask to generate a large amount of intensity data. Because the pattern formed by the mask is known, image processing techniques can be used to extract useful information from the raw image.

1.2.2 Digital Sun Sensors

All digital sun sensors utilize the same basic operational principles. Incoming light from the sun strikes a sensor mask, which will permit some desired light pattern through. The mask may be inscribed with slits or holes, depending on which form of light is most useful for the sensor type. The light that passes through the mask will then fall on a detector, and will be converted to an electrical signal. This constitutes the raw image data. Processing algorithms will then be employed which will translate this signal into an attitude reading.

Whereas analog sensors use a small number of photosensitive elements, each individually responding to the light from a large portion of the field-of-view, digital sun sensors utilize an array of pixels, each responsible for a

1. INTRODUCTION

very small portion of the field of view. As the incoming lights falls across a number of these pixels, the overall shape of the intensity data can be very well represented when discretized across a larger number of elements. Further, the smaller size of these pixels allows for better separation between light from the sun and stray light that falls across the entire array. Because a larger amount of intensity data are returned, digital sensors generally require more data space and more processing than their analog counterparts.

The image processing is generally comprised of two steps. First, the raw image data are distilled into a small number of important parameters. These parameters, used with the sensor and mounting geometry, are then used to determine the originating sun vector that caused the image. The mathematics required in the latter step are well understood, given some defined geometric sensor model. Improving the algorithms that process the raw data and better identification of the sensor calibration parameters comprise a much more interesting problem.

1.2.3 Digital Sun Sensor Improvement

Digital sun sensors utilize a large number of small pixel elements to give intensity data. Although resolution of the original signal is fairly high, extremely accurate identification of parameters that identify the original signal is desired. Processing algorithms are often susceptible to image noise and calibration errors.

A number of different algorithms are available that can process the raw image data. These differ in accuracy, both in common image cases and in which 'special cases' the accuracy may be reduced or the algorithm may fail entirely. Further, depending on the complexity of the algorithm used, the amount of time needed by hardware to process the image may vary.

Reducing calibration errors requires a very detailed understanding of the sensor geometry. Further, the errors that can be introduced from sensor mounting must also be accounted for. The calibration procedure is only

effective when all contributors are identified and correctly represented mathematically. Better representation of the sensor calibration parameters can then improve overall performance when the sensor is in use.

1.3 Work Objectives

The primary objective of this thesis is to derive methods that allow a better understanding of digital sun sensor behaviour. New methods will be presented that will better represent the underlying sensor behaviour. Further, a new way of using the sensor data will be explored that may result in more accurate observations. The thesis will focus on the following objectives:

1. Modelling the behaviour of digital sun sensors. Correct identification of device behaviour yields insights into the limitations of the digital sun sensors, and provides tools which can be used to simulate device behaviour in the future.
2. Identifying important calibration parameters. This involves both identification of the parameters which may lead to calibration error and correctly modelling the parameter.
3. Improving sun sensor calibration. Continuous evaluation of calibration procedures and strategies that incorporates new parameters and sensor models can lead to better performance.

Here, the objective is not to define the underlying physics that cause a given image, but instead will rely on empirical models to describe how the sensor will be expected to behave, namely, data sets taken using the sensor will provide the insights into the sensor behaviour.

This paper will explore modelling and simulation of the SS-256 and SS-411 class sensors from Sinclair Interplanetary. By improving and implementing calibration parameters, we hope to improve sensor performance. Modelling

1. INTRODUCTION

the sensor behaviour yields insights into limitations of the sensor, and also gives some benchmarks that can be used to test against in the future.

1.3.1 Thesis Outline

Background information that leads to the investigations mentioned will be presented in Chapter 2. This will include a description of the sensors used in the investigation, as well as a survey of the algorithms used to translate image data into observation results. Changes to the way the peaks are modelled, namely, a new representation of peak width, will be discussed in Chapter 3. A method for defining how well a sensor performs is derived and implemented in Chapter 4. A new method of taking sensor observations is described in Chapter 5. This method will be conducted using both an arbitrary sensor for which specific error data are unknown and a sensor for which error data are intimately known. Finally, Chapter 6 will conclude the research and provide some direction for further research on the topic.

Chapter 2

Background

Most of the previous work regarding sun sensors involves two important tasks. First, there is a desire to improve accuracy in the information derived from a sensor image. Second, more accurate calibration is desired so that the physical properties of the sun-sensor can be accurately described. Inaccuracies in either of these areas can contribute to error when the sun-sensor is in use.

In order to fully understand the previous work done in these areas, one must first define how the sun-sensor interprets the incoming sun image and how this information is displayed. Then, methods that translate that sensor data into a sun-vector must be explored. Once these basics are known, different methods of varying accuracy will be explored that allow us to find the underlying image data, and different ways of representing sensor unknowns will be presented.

2.1 Sun Sensor Optics

The mathematical models presented here are designed using the physical parameters from the Sinclair Interplanetary SS-256 and SS-411 digital sun sensors. This is a two-axis digital sun sensor that has a 70-degree arc field-of-view. The sensor mask contains two pairs of orthogonal slits. The detector

2. BACKGROUND

array is a 16 mm-long linear array of 256 pixels, oriented at approximately 45-degrees to the slits in the mask.

In order to calculate the incoming sun vector from the displayed image, the position of the image array on the image plane needs to be accurately described. Fig.(2.2) describes the image plane geometry.

The sensor works by restricting the incoming light that is able to hit the image array via the slits. Because only the light that passes through the slits reaches the array plane, there will be four strips of light that reach the image plane. Where these intersect with the array itself, they will be resolved into 'peaks' of light. A graphical representation of the mask and image plane is shown in Fig.(2.1).

An ideal representation of the image array is shown in Fig.(2.2), however the actual image array is subdivided into two 128-pixel length arrays, placed lengthwise beside each other. These two arrays have a small gap between them, and the individual pixels near the gap are slightly warped. Fig.(2.3) shows the two arrays side by side, with the slight pixel warping that occurs near the gap.

Once peak locations are known, ray tracing can be used to calculate the incident sun vector. This is done using the internal geometry of the sensor, including the relative offset between the array plane and sensor mask, as well as the index of refraction. The originating sun-vector with respect to the platform frame [Enright, 2008] \hat{s}_0 , and the sun-vector with respect to the sensor frame, \hat{s} are related through the sensor parameters. Given the definition of the incident sun-vector

$$\hat{s}_0 = \begin{bmatrix} s_{0,x} \\ s_{0,y} \\ s_{0,z} \end{bmatrix} \quad (2.1)$$

The components of \hat{s} are found individually via:

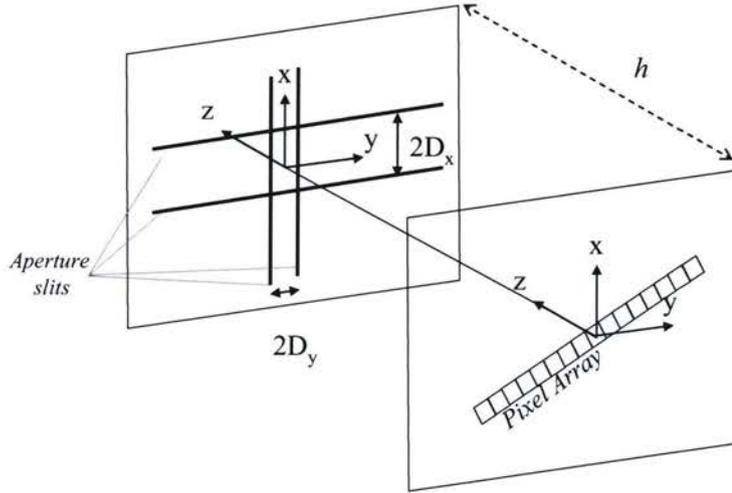


Figure 2.1: Geometric Model of Sensor Showing Aperture and Image Planes.[Enright, 2008]

$$\begin{aligned}
 s_x &= \frac{1}{n_{glass}} (s_{0,x} \cos \phi + s_{0,y} \sin \phi) \\
 s_y &= \frac{1}{n_{glass}} (-s_{0,x} \sin \phi + s_{0,y} \cos \phi) \\
 s_z &= -\sqrt{1 - s_x^2 - s_y^2}
 \end{aligned} \tag{2.2}$$

Here, the angle ϕ corresponds to a right-handed rotation about the positive z-axis between the sensor and mounting platform.

Now given the distance between slits of $2D_x$ and $2D_y$ for the x- and y-axis slits, respectively, and the distance h between the mask and the center of the image plane, the position of the peaks along the array can be derived

2. BACKGROUND

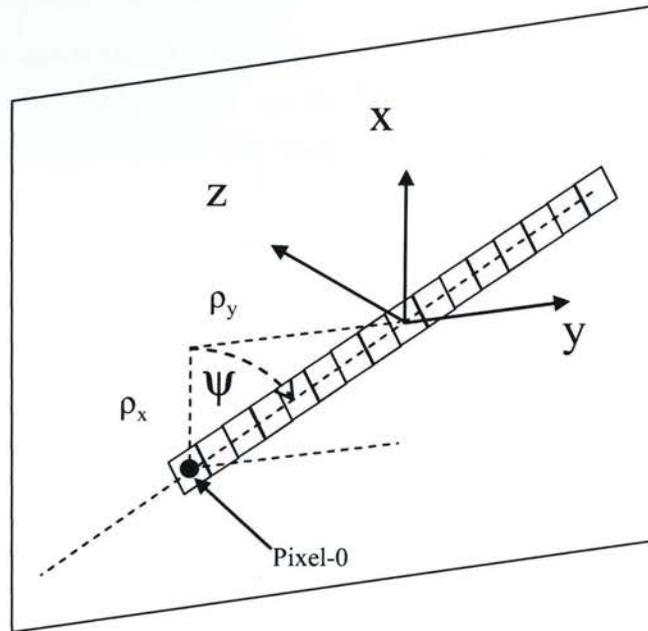


Figure 2.2: Geometric Model of the Image Plane.

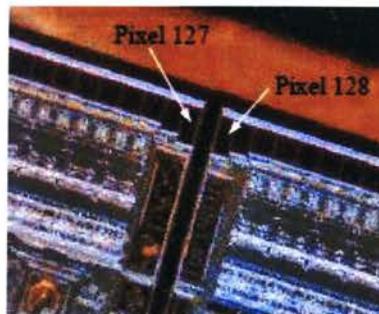


Figure 2.3: Image of the Central Array Gap.[Enright, 2008]

2.2. Identifying the Image Characteristics

as [Enright, 2008]:

$$m_{x1} = \frac{1}{\gamma \cos \psi} \left[- \left(\frac{s_x h + s_z D_x}{s_z} \right) - \rho_x \right], m_{x2} = \frac{1}{\gamma \cos \psi} \left[- \left(\frac{s_x h - s_z D_x}{s_z} \right) - \rho_x \right] \quad (2.3)$$

$$m_{y1} = \frac{1}{\gamma \sin \psi} \left[- \left(\frac{s_x h + s_z D_y}{s_z} \right) - \rho_y \right], m_{y2} = \frac{1}{\gamma \sin \psi} \left[- \left(\frac{s_x h - s_z D_y}{s_z} \right) - \rho_y \right] \quad (2.4)$$

Although this does relate the sun vector and the peak position on the image array, typical sun sensor operation requires that the inverse of these relationships must be computed. This can be done by writing the temporary quantities [Enright, 2008]:

$$A_m = \left[\Delta T \cos \psi \left(\frac{m_{x1} + m_{x2}}{2} \right) + \rho_x \right] \quad (2.5)$$

$$B_m = \left[\Delta T \sin \psi \left(\frac{m_{y1} + m_{y2}}{2} \right) + \rho_y \right] \quad (2.6)$$

When these are expanded using the definitions in Eq(2.3) and Eq(2.4), the following relation can be found:

$$s_z = - \sqrt{\frac{h^2}{A_m^2 + B_m^2 + h^2}} \quad (2.7)$$

The other two components of \hat{s} are then:

$$s_x = - \frac{A_m}{h} s_z \quad (2.8)$$

$$s_y = - \frac{B_m}{h} s_z \quad (2.9)$$

2.2 Identifying the Image Characteristics

Light incident on the pixel array varies in a spatially continuous pattern across the detector. Peaks resulting from direct light through the mask slits are expected to correspond to the high-intensity values on the array, and other areas on the array should receive very little light. However, there will likely be some noise from stray light, noise from internal reflection, and noise from the array itself, all contributing to some variation in the recognized intensity.

Because the mask has four slits creating bands of light on the image plane, the expectation is that there will be either three or four separate peaks that will manifest on the image array. Three peaks will show if there is a light-band overlap that occurs on the image array, otherwise there will be four peaks. An example of these two cases is shown in Fig.(2.4) On the return of image data, the locations of these peaks are found on the image array. At this point, only the locations of these peaks are known, but not which mask slit as contributed to which peak. The peaks as seen on the image array in absence of this extra information are called *physical peaks*. After the peaks on the image have been found, they need to be matched with an originating slit in the mask. The slits themselves produce *logical peaks* based on the sun-vector direction. By mapping the logical peaks onto the physical peaks, the calculations in Eq(2.5) and Eq(2.6) can be performed.

It is not enough simply to find the maximum values across the array as this results in very low accuracy. We would like to find the peak locations to subpixel accuracy. However, because the image peaks are sharply defined, simple methods can be adopted which will allow for mapping between the physical and logical peaks. Many of the subpixel techniques that will be described require that a successful map between peaks has already been done.

In order to detect maxima in the image array, a 5-pixel window is used and slides across the array. Here, the central pixel is compared to the two on

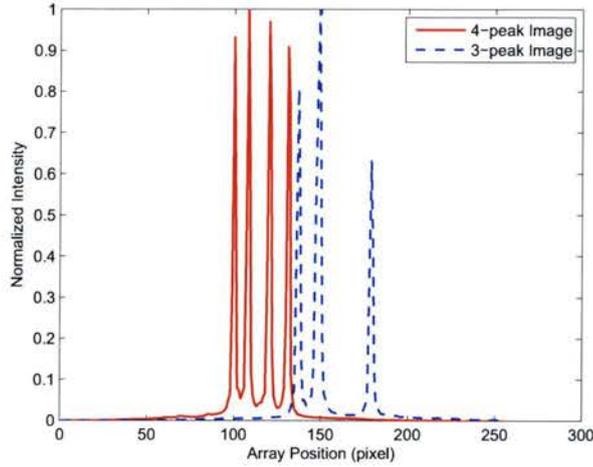


Figure 2.4: Four-Peak versus Three-Peak Image.

either side to see if it is a local maximum. The following relation will define where the peak maxima are:

$$I[n] > I[n - 2], I[n - 1], I[n + 1], I[n + 2] \quad (2.10)$$

Doing this creates a set of *physical* peaks located at points n_i along the image array. At this point, these should be associated with contributions of particular slits in the mask - the *logical* peaks. The physical peak locations found thus far can be stored in a vector P :

$$P = \{\tilde{m}_1, \tilde{m}_2, \dots\} \quad (2.11)$$

In most cases, P is of length 4, since most sun-vectors result in a 4-peak image. In the overlap case, P will be of length 3, and one of those three detected peaks will correspond to two logical peaks.

The expected spacing resulting from corresponding slit pairs, $2L_x$ and $2L_y$, can be pre-calculated from the sensor geometry. Recalling that the light-bands from the x- and y-axis slits will be perpendicular to those axes

2. BACKGROUND

as shown in Fig.(2.2), the peak separation as seen by the image array can be calculated by accounting for the angle ψ :

$$L_x = \frac{\Delta T}{2} (m_{x2} - m_{x1}) = \frac{D_x}{\cos \psi} \quad (2.12)$$

$$L_y = \frac{\Delta T}{2} (m_{y2} - m_{y1}) = \frac{D_y}{\cos \psi} \quad (2.13)$$

An error term can now be defined, Φ , which will describe the error in matching logical peaks to the physical peaks detected on the array. This can be described mathematically as:

$$\Phi = \left\{ L_x - \frac{\Delta T}{2} (\tilde{m}_{x2} - \tilde{m}_{x1}) \right\}^2 + \left\{ L_y - \frac{\Delta T}{2} (\tilde{m}_{y2} - \tilde{m}_{y1}) \right\}^2 \quad (2.14)$$

By rearranging the peak locations in P and substituting into the relation for Φ , errors can be obtained for the assumed peak matching. The arrangement of peak locations m that minimize the error Φ will correspond to the correct peak matching.

Although this will yield the logical peak location to the nearest pixel, there is a strong desire to improve this to sub-pixel accuracy. There are a number of algorithms that can accomplish this to varying degrees of success.

2.3 Super-Resolution

Super-resolution techniques are commonly used when attempting to yield information from digital sensors. These techniques offer a way of taking the raw image data received and obtaining subpixel estimates. For the sun-sensor example, subpixel techniques give the ability to detect the position of the maximum of each peak shape to subpixel accuracy. These techniques are extremely important in pulling accurate attitude data from digital sensors.

2.3.1 First-Order Centroiding

First order centroiding is a common method of finding the subpixel point location of an extended light source. This method is used extensively with star-trackers, where a point source of light is deliberately defocussed such that the light gives a 2-d peak shape when it reaches the image plane.

The methods behind this technique are described by Liebe[Liebe, 2002] and Rufino[Rufino and Accardo, 2003]. Given a matrix of intensities $I(x, y)$, a function of the location (x, y) on the image array, the centroid can be found by performing the following operations:

$$x_{cm} = \frac{\sum_x \sum_y x I(x, y)}{\sum_x \sum_y I(x, y)} \quad (2.15)$$

$$y_{cm} = \frac{\sum_x \sum_y y I(x, y)}{\sum_x \sum_y I(x, y)} \quad (2.16)$$

Typically, the summations are bounded by a region of interest (ROI) that contains all of the image data[Rufino and Accardo, 2003].

Because this method does not consider the originating image shape or require any additional input beyond the image itself, it can be performed with no previous knowledge of the illumination pattern.

2.3.2 Non-Linear Least Squares

The non-linear least squares (NLSQ) method involves fitting the image data to a parameterized model of expected illumination. In this case, the peak location is the parameter that is of most interest.

The NLSQ model will be modified from a Gaussian distribution model of the peak shape [Enright, 2008]. This simplified model will reflect the shape of the peaks as seen in the image. Given a parameterized equation I_{mdl} that

2. BACKGROUND

can approximate the shape of one of the peaks, we will build a method by which to describe the whole image.

First define the individual peak shape via the equation

$$I_{mdl_0}(\tau) = e^{-\frac{\tau^2}{2\sigma^2}} \quad (2.17)$$

In this equation, τ is a coordinate which maximizes the value of I_{mdl_0} at $\tau = 0$. The value of σ^2 is closely tied to the width of the peak and will be determined outside the error minimization.

Because four peaks are expected across the image array, the illumination pattern will be defined as follows:

$$I_{mdl} = a_1A(\tau) + a_2B(\tau) + a_3C(\tau) + a_4D(\tau) \quad (2.18)$$

We will offset the model equation I_{mdl_0} to match the positions of the peaks on the image array. This will result in the following definitions:

$$\begin{aligned} A &= I_{mdl_0}(\tau - m_{x1}) \\ B &= I_{mdl_0}(\tau - m_{x1} - \frac{2L_x}{\Delta T}) \\ C &= I_{mdl_0}(\tau - m_{y1}) \\ D &= I_{mdl_0}(\tau - m_{y1} - \frac{2L_y}{\Delta T}) \end{aligned} \quad (2.19)$$

The value of $\frac{2L_i}{\Delta T}$ is a measure of the distance between slits on the sensor mask. We expect this to be equal to the distance between the peaks that those slits create on the image array. We only have six parameters that must be fit to the image. Arranged in a vector, these are

$$\lambda = \left[a_1 \quad a_2 \quad a_3 \quad a_4 \quad m_{x1} \quad m_{y1} \right]^T \quad (2.20)$$

The scaling parameters a_1 through a_4 describe the amplitude of the peaks in the image. The parameters m_{x1} and m_{y1} describe the position of the peaks along the image array. The latter are of most interest to us, however we must solve for the former parameters as well.

In order to find the vector λ , the error between the actual and modelled image must first be found via:

$$d\beta_n = I[n] - I_{mdl}(\tau_n, a_1, a_2, a_3, a_4, m_{x1}, m_{x2}) \quad (2.21)$$

Taking the partial derivatives of the image model with respect to the model parameters yields the following[Enright, 2008]:

$$H_\lambda = \begin{bmatrix} \frac{\partial I_{mdl}}{\partial a_1} |_{\tau_1} & \frac{\partial I_{mdl}}{\partial a_2} |_{\tau_1} & \dots \\ \frac{\partial I_{mdl}}{\partial a_1} |_{\tau_2} & \ddots & \\ \vdots & & \frac{\partial I_{mdl}}{\partial m_{x2}} |_{\tau_N} \end{bmatrix} \quad (2.22)$$

The elements of H_λ are calculated using the current value of the model parameters, and the array position τ . From the definition of H_λ , the values of Eq(2.19) become evident:

$$\begin{aligned} \frac{\partial I_{mdl}}{\partial a_1} |_{\tau} &= A(\tau) \\ \frac{\partial I_{mdl}}{\partial a_2} |_{\tau} &= B(\tau) \\ \frac{\partial I_{mdl}}{\partial b_1} |_{\tau} &= C(\tau) \\ \frac{\partial I_{mdl}}{\partial b_2} |_{\tau} &= D(\tau) \end{aligned} \quad (2.23)$$

The remaining two terms are [Enright, 2008]:

$$\frac{\partial I_{mdl}}{\partial m_{x1}} |_{\tau} = - \left[a_1 \left(\frac{\partial}{\partial u} I_{mdl}(u) |_{u=\tau-m_{x1}} \right) + a_2 \left(\frac{\partial}{\partial u} I_{mdl}(u) |_{u=\tau-m_{x1}-\frac{2L_x}{\Delta T}} \right) \right] = a_1 E(\tau) + a_2 F(\tau) \quad (2.24)$$

$$\frac{\partial I_{mdl}}{\partial m_{y1}} |_{\tau} = - \left[b_1 \left(\frac{\partial}{\partial u} I_{mdl}(u) |_{u=\tau-m_{y1}} \right) + b_2 \left(\frac{\partial}{\partial u} I_{mdl}(u) |_{u=\tau-m_{y1}-\frac{2L_y}{\Delta T}} \right) \right] = b_1 G(\tau) + b_2 H(\tau) \quad (2.25)$$

This simplifies the definition of H_λ , which can now be simply written as:

$$H_\lambda = \left[A \quad B \quad C \quad D \quad (a_1E + a_2F) \quad (b_1G + b_2H) \right] \quad (2.26)$$

For each iteration, the parameter update step $d\lambda$ can be shown as [Enright, 2008]:

$$H_\lambda^T d\beta = (H_\lambda^T H_\lambda) d\lambda \quad (2.27)$$

2.3.3 Linear-Phase

One method of estimating the peak positions involves using phase-correlation. Given a discrete image $I_0[n]$, its frequency domain representation can be obtained by performing a discrete Fourier transform (DFT) [Godard and Enright, 2006]:

$$I[n] \stackrel{DFT}{\Leftrightarrow} S[k] \quad (2.28)$$

Also used is a DFT property regarding space-domain shifts and their representation in the frequency domain [Mitra, 2006]:

$$I[n - \alpha] \stackrel{DFT}{\Leftrightarrow} S[k] e^{-j \frac{2\pi k \alpha}{N}} \quad (2.29)$$

Here, the illumination pattern has been shifted by α samples, which is equivalent in the frequency domain to multiplication by a linear-phase term [Enright, 2008].

By defining a reference DFT $S_0[n]$ corresponding to a known zero-shift image, it is possible to isolate the exponential term. Given the DFT of an actual shifted image $S_\alpha[n]$, this extraction can be done through the relation [Godard and Enright, 2006]

$$\Psi[k] = \angle \left(\frac{S_\alpha[n]}{S_0[n]} \right) = \angle \left(e^{-j \frac{2\pi k \alpha}{N}} \right) = -\frac{2\pi k \alpha}{N} \quad (2.30)$$

A least-squares fine line of $\Psi[k]$ gives us an estimate of the slope term, $\tilde{\alpha}$.

This is related to the peak position estimate \tilde{m} via

$$\tilde{m} = \pm \frac{L}{2} + \tilde{\alpha} \Delta T \quad (2.31)$$

In order to perform the linear-phase algorithm, the contributions from the x and y slits need to be separated, since the shift of each pair will be found separately using their own reference images, i.e., the image will be considered as a superposition of two sub-images from each peak pair.

$$I[n] = I_x[n] + I_y[n] + I_{noise}[n] \quad (2.32)$$

Here, I_x and I_y are the contributions from the pairs of slits, and will be used to find the shifts via linear-phase. I_n is the image noise. Further, note that I_x and I_y are shifted reference images. They can be described as:

$$I_x[n] = I_{x0}[n - \alpha_x] \quad (2.33)$$

$$I_y[n] = I_{y0}[n - \alpha_y] \quad (2.34)$$

Given an image, the linear-phase algorithm decomposes the image down into two separate sub-images corresponding to contributions by both the x - and y -axis sets of peaks [Foroosh, 2002]. It does this by estimating the positions of individual peaks on the image array. Because the separation between each peak pair is known, the image can be decomposed into the two sub-images by considering the approximate locations.

Some difficulty arises when there is a superposition of peaks. In this instance, the overlapping peak location is considered belonging to both subsets of images. However, we then use the other non-overlapping peaks to perform the linear-phase algorithm.

2.4 Algorithm Testing and Sensor Calibration

At this point, we need to develop a number of procedures that both test the accuracy of the NLSQ and linear-phase algorithms, as well as calibrates the sun-sensor. A laboratory setup must be devised that simulates the exposure of the sensor to sunlight. Also, a method of finding the calibration parameters using the actual data needs to be found. This yields a method of both verifying the ability of developed algorithms to accurately predict the sun position and creating a repository of images that can be used to test different hybrids and modifications to see if performance can be improved.

2.4.1 Calibration Approach

A calibration approach was developed that enables us to continually update our sensor parameterization and improve our sensor performance. Our approach is comprised of five steps:

1. Examine sensor operation.
2. Develop sensor behaviour models based on parameters regarding sensor or lab setup.
3. Calibrate the parameters based on a suite of test images and associated truth model. Allow parameters to fluctuate as we minimize error between estimates and truth.
4. Examine residual error for structure. Systematic error (any non-white noise error) is suggestive of unmodelled or poorly modelled system behaviour.
5. Repeat.

These steps are performed until an acceptable performance level has been reached.

2.4.2 The Sun Sensor Test Setup

A laboratory setup has been devised that can test for sensor accuracy. The sun-sensor itself is mounted onto a three-axis rotary platform. The rotary platform is computer-controlled and can be instructed to move to different angular positions on all three axes in 0.001-degree increments. Mounting offsets can be handled through a calibration procedure, which simplifies the mounting process.

The sun is simulated with a xenon arc-lamp. The projected light is constrained via an adjustable iris, which is adjusted such that the lamp appears to have the same angular diameter as the sun. Fig.(2.5) depicts the setup currently used for testing.

2.4.3 Calibration Image Sets

Although physical dimensions for the sun-sensor are often given by the manufacturer, there is an element of uncertainty associated with the actual sensor – often due to manufacturing errors and limitations. Uncertainty in physical parameters can have large error effects in identifying the true orientation of the spacecraft [Enright, 2007]. Thus, we would like to measure these parameters accurately so that, when confronted with an image, we can minimize the error contribution due to incorrect parameterization.

In order to perform the calibration, we need to all of the parameters we would like to investigate. We then capture a number of images corresponding to various positions in the field-of-view (FOV) of the sensor. Since we define the relative position of the sensor with respect to the incoming sunlight, we know what the true sun vector should be. We compare this to the sun vectors as determined by the inverse sensor model to find the difference, or error, between the two.

A set of test images must be generated in order to perform the calibration. In most cases, approximately 400 points throughout the FOV were used for

2. BACKGROUND

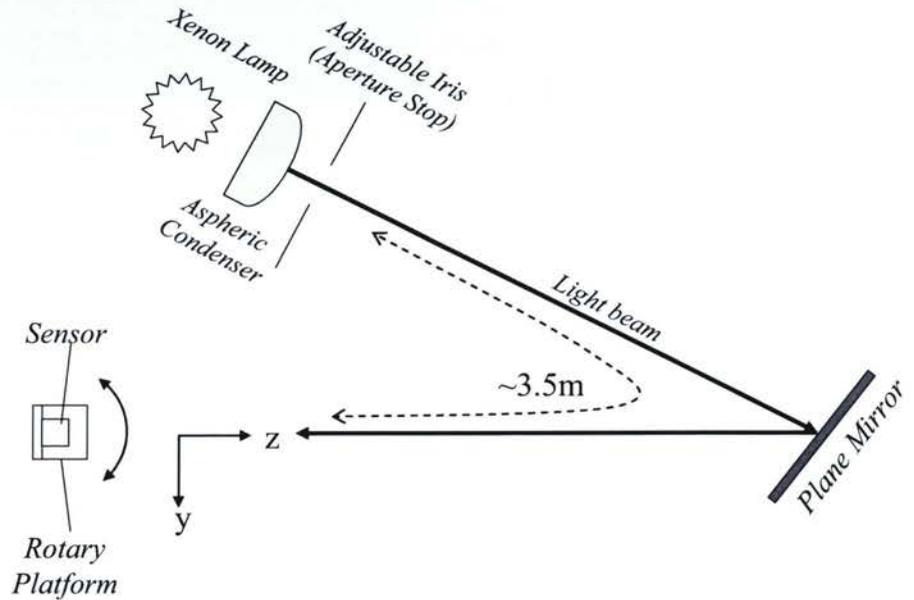


Figure 2.5: Experimental Setup.

calibration, although in some cases 800 points were used. An algorithm was developed so that the desired number of points would be distributed amongst a number of user-defined annuli, such that the sampling across the field of view was relatively even.

To do this, consider the total area subtended by the sensor FOV (in

steradians). This can be calculated from

$$\Omega_{tot} = 2\pi (1 - \cos \zeta_{max}) \quad (2.35)$$

Here, the value of ζ_{max} is the maximum offset of the image set from the boresight – usually the FOV limit of 70-degrees. We then divide the total area into m annuli, the value of which is defined by the user. The outer annulus lies at ζ_{max} and an additional sample is taken at $\zeta = 0$. Given that the user has defined m desired annuli, the algorithm subdivided the field-of-view into a number of circular 'ribbons' with the width of those ribbons remaining constant. Mathematically, the following relation was satisfied:

$$\frac{1}{2} (\zeta_{outer} - \zeta_{inner}) + m (\zeta_{outer} - \zeta_{inner}) = \zeta_{max} \quad (2.36)$$

Rewritten so that the ribbon width can be explicitly found, this becomes

$$\zeta_{outer} - \zeta_{inner} = \frac{\zeta_{max}}{\frac{1}{2} + m} \quad (2.37)$$

Because the field of view contains M distributed points, each individual ribbon contains a percentage of those points defined by the ratio of its area to the total area covered by the image set. Given that each ribbon covers an area Ω_i , then

$$\Omega_i = 2\pi (\cos \zeta_{i,inner} - \cos \zeta_{i,outer}) \quad (2.38)$$

Each individual annulus contains a percentage of the total defined points equivalent to the percentage of the total area occupied by the annulus, i.e.,

$$M_i = \frac{\Omega_i}{\Omega_{tot}} M_{tot} = \frac{\cos \zeta_{i,inner} - \cos \zeta_{i,outer}}{1 - \cos \zeta_{max}} M_{tot} \quad (2.39)$$

2.4.4 Calibration Procedure

In order to calibrate the sensor parameters so that they are an accurate representation of the physical sensor, we use a minimization routine with regards to the error between the two sets of sun vectors. For each true sun-vector and it's corresponding estimate, we calculate the angle θ_{err} between the two vectors via:

$$\theta_{err} = \arccos\left(\frac{\hat{s}_{true} \cdot \hat{s}_{est}}{\|\hat{s}_{true}\| \|\hat{s}_{est}\|}\right) \quad (2.40)$$

Given an initial guess of the vector of parameters, we define the overall error as follows:

$$y_{err} = \sum |\theta_{err}| \quad (2.41)$$

We then use a Matlab built-in minimization routine to find the minimum value of this error, and return those parameters which generate this minimum value.

From the previous parameter definitions regarding the sensor geometry, there are seven parameters which need to be found through calibration. Combined with several others that are fixed, the sensor physical geometry can be adequately described using the values depicted in Table(2.1).

There are also a number of parameters which describe the mounting of the sensor to the rotary platform, and describe offsets in the interaction between the xenon lamp sun-simulator and the zero-position of the mounted sensor. Unlike the sensor-physical parameters, the mounting parameters need to be recalculated every time the sensor is re-mounted to the platform. The mounting offset that results can be described using the following equation[Enright, 2008]:

$$\vec{r}_{sun} = R_x(\eta_1) R_y(\eta_2) (K^{-1}\vec{r}_1 + \vec{r}_2) \quad (2.42)$$

Table 2.1: Physical Sensor Paramaterization

Parameter	Description	Unit	Type	Assumed Value
ϕ	Array-Mask Offset	radian	Calibration	0
ψ	Image Array Angle	radian	Calibration	$\pi/4$
ρ_x	Pixel-0 x-location	metre	Calibration	-5.7×10^{-3}
ρ_y	Pixel-0 y-location	metre	Calibration	-5.7×10^{-3}
h	Array-Mask Dist.	metre	Calibration	4.0×10^{-3}
D_x	Half-Slit width in x	metre	Fixed	7.5×10^{-4}
D_y	Half-slit width in y	metre	Fixed	2.5×10^{-4}
n_{glass}	Mask Refraction	n/a	Fixed	1.51
γ	Pixel Width (aka ΔT)	metre	Fixed	63.5×10^{-6}

Here, η_1 and η_2 describe the angular offset (or tip/tilt offset) of the sensor boresight mounting with respect to normal. \vec{r}_1 is a vector from the centre-of-rotation of the rotary platform to the sun (the xenon sun-lamp). \vec{r}_2 is a vector from the centre-of-rotation of the rotary platform the the sensor origin. K itself is a transformation matrix based on the sensor platform rotation kinematics. Because the sun-vector is normalized as \vec{s}_0 , the scale of Eq(2.42) can be ignored, simplifying the calibration process. Here, $r_{1,x}$ is fixed at unity and the other values are calibrated around this.

2. BACKGROUND

Chapter 3

Peak Width Correction

An integral part of being able to identify the sun vector is to be able to predict the width of the peaks as they appear on the array. This is important when predicting the overall shape of a logical peak, when comparison with a physical peak is required. Previously, during the implementation of the NLSQ algorithm, the peak width of all logical peaks was assumed constant, irrespective of the array position of those peaks.

In order to investigate how peak width varies across the image array, a method of collecting peak width data was required, and a method of prediction was needed based on array position. As the investigation was conducted, a clear correlation between peak width and array position was found.

3.1 Peak Shape Theory and Data Collection

3.1.1 Theory Development

Describing the peak shape requires that an equation be derived that can be used as a model. A Gaussian model was used to fit to the image peaks, similar to the implementation used in the NLSQ algorithm. Here, the mathematical model used will be:

3. PEAK WIDTH CORRECTION

$$I_{mdl} [k] = Ae^{-\frac{(k-\bar{k})^2}{2\sigma^2}} \quad (3.1)$$

In the above equation, A is a scaling factor that will represent the amplitude of the peak; \bar{k} is the central value of the peak-shape; the value of σ represents the peak width. In this equation, all of these parameters are unknown. In the local sense, all of these parameters will need to be found for each individual peak.

The parameter set will then be rendered as:

$$P = \{\sigma, \bar{k}, A\} \quad (3.2)$$

To do this, an error between the image as modelled by the parameter set and the true image is defined. This error is written as:

$$\varepsilon = \sum_k (I[k] - I_{mdl} [k])^2 = \sum_k \left(I[k] - Ae^{-\frac{(k-\bar{k})^2}{2\sigma^2}} \right) \quad (3.3)$$

Values of the parameter set P are then chosen such that this error is minimized.

3.1.2 Width Data Collection

To examine how the width of the image peaks varies across the image array, a calibration image was taken. For each of these images, the physical peaks were associated with slits in the mask, so that the peak width variation could be associated directly with the peaks cast by each slit. Overlapping peaks were removed from the set to improve fitting accuracy.

For each non-overlapping peak, an individual curve based on the Gaussian shape was fitted to the image data. An example of this fit done to the curve in Fig.(??) is shown in Fig.(3.1). When the values of \bar{k} and σ were compared from the fit of the model in Eq(3.1), trends emerged in the data.

For each of the images in the calibration set, the location and peak width

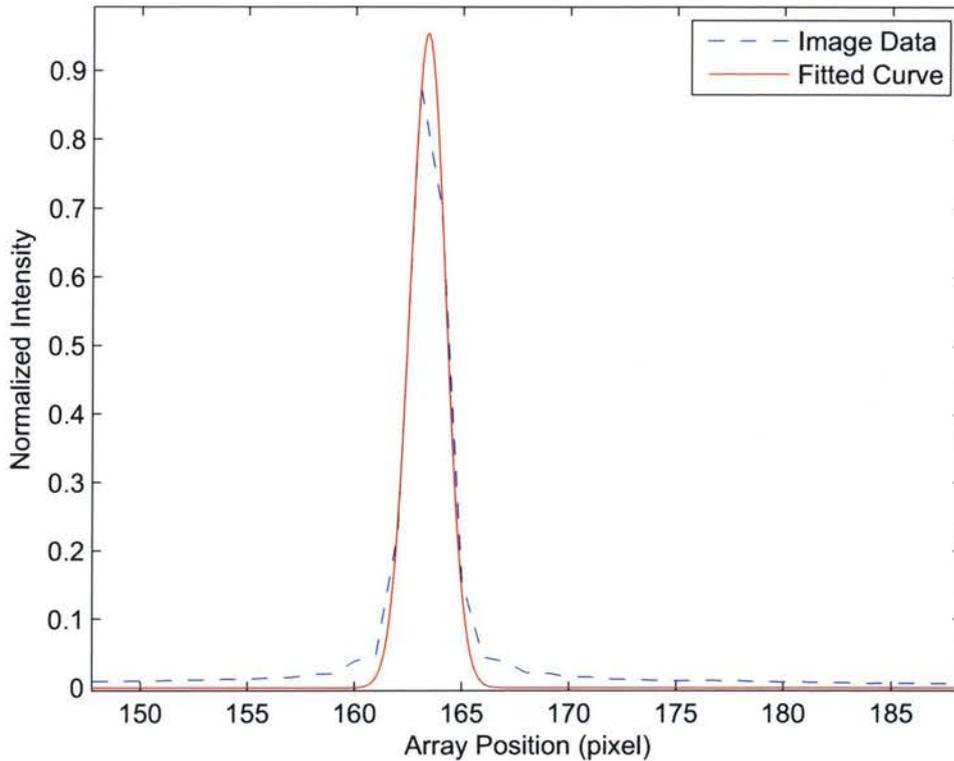


Figure 3.1: Image Peak with Curve Fit.

was found for each logical peak. When the peak width was plotted against the location, the size of the peak width term seemed to follow an even function. An example of this behaviour can be seen in the raw data from a single logical peak in Fig.(3.2). This indicated that better performance may be acquired by allowing the algorithm to vary the peak shape in accordance with this data. For each sensor then, a curve fit was which described peak width as a function of array location was found.

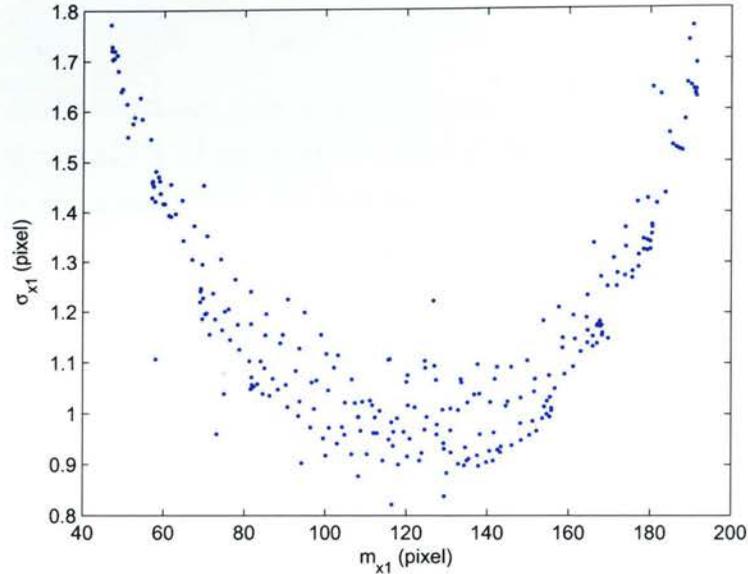


Figure 3.2: Fitted Peak Width by Array Position.

3.2 Relating Peak Width to Array Position

3.2.1 Fit Equation Development

Given that the array position has a clear effect on the peak width, a method of describing how these two quantities relate was needed. In this case, a theoretical development of this relationship was not desired, and instead empirical techniques were employed.

From trial and error, it was found that an even, fourth-order polynomial was closest to the true behaviour of the peak widths with respect to the array position. It was decided that the trend would be described via the equation:

$$\sigma_{fit} = A(x - x_0)^4 + B(x - x_0)^2 + C \quad (3.4)$$

Here, x describes the array position, x_0 is the position of the minimum value of the curve, and A through C are scaling constants to be fitted. After

3.2. Relating Peak Width to Array Position

doing peak fits for individual peaks in all the images in the set and all peak widths cast by a single logical peak were compared to the array position, a fit was done so that the peak width could later be predicted based on the array position. To do the minimization, the parameter set was defined as the scaling constants and the minimum value position, i.e., the parameter set was

$$P = \{A, B, C, x_0\} \quad (3.5)$$

To perform the minimization, an error function needed to be defined between the actual peak width data and the function that is needed to describe it. Again, a squared-error approach was used, entailing the following error function:

$$\varepsilon = \sum_i (\sigma_{fit_i} - \sigma_{true_i})^2 = \sum_i \{ [A(x_i - x_0)^4 + B(x_i - x_0)^2 + C] - \sigma_{true_i} \} \quad (3.6)$$

This error function was minimized to find the parameter set P . This curve fit was performed twice – one for each of the x- and y-axis pairs of peaks. We will treat the corresponding second slit of a logical pair as a constant shift from the first, as was done in Eq(2.19). Mathematically, the following relation is expected:

$$\sigma_{theo,x_1}(k, k_0) = \sigma_{theo,x_2}(k, k_0 + 2L_x) \quad (3.7)$$

$$\sigma_{theo,y_1}(k, k_0) = \sigma_{theo,x_2}(k, k_0 + 2L_x) \quad (3.8)$$

Since we only need to perform two curve fits, only two sets of parameters for two of the logical peaks need to be passed to future algorithms.

3. PEAK WIDTH CORRECTION

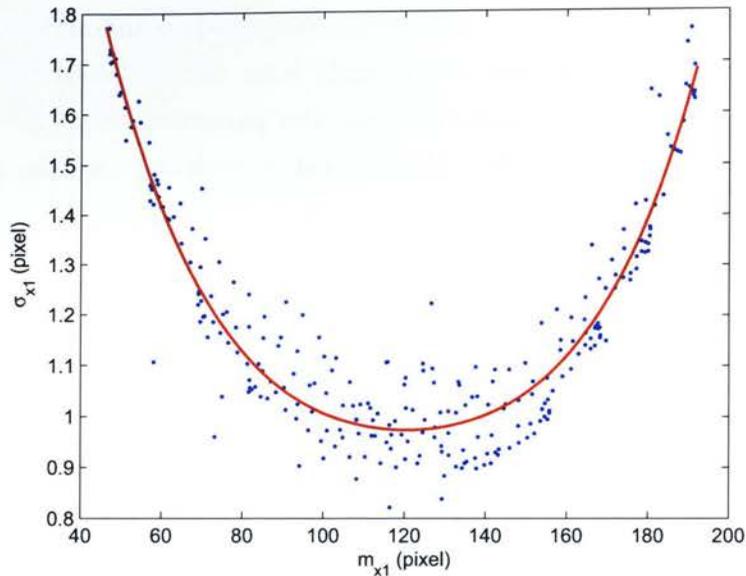


Figure 3.3: Peak Width by Array Position with Curve Fit.

3.2.2 Curve Fit Results

Given the image data, values of peak width as well as the peak position was found. An example of this was shown previously in Fig.(3.2). Given the example peak width set in Fig.(3.2), a curve was fitted to the data. The results for each parameter are summarized in Table(3.1). Plotting the curve in Eq3.4 with these values yields Fig.(3.3).

Table 3.1: Fourth-Order Curve Fit Parameters

Parameter	Value
A	1.30×10^{-8}
B	7.24×10^{-5}
C	0.974
x_0	120.42

3.3 Calibration Revision

A number of revisions were made to the calibration parameters presented in Table(2.1). During the width correction work, many revisions were made to the initial parameter set. These changes will be outlined here.

3.3.1 Index of Refraction

In the initial parameter set, the index of refraction of the mask was assumed to be 1.51. This was converted to part of the calibration parameter set because it is crucial in relating peak position to incident sun-vectors.

If the index of refraction is incorrect, the change in direction of the incident light passing through the mask will be incorrectly calculated in Eq(2.2). This can be a significant error when we are attempting to convert peak positions on the image array to incident sun vectors.

3.3.2 Longitudinal Gap

The initial parameter set treats the array as a contiguous line of 256 pixels. The array itself is actually comprised of two linear arrays of 128 pixels placed end-to-end, and contains a small gap in between the two. This was shown previously in Fig.(2.3).

The calibration set was updated to include a longitudinal gap term. This parameter defines the distance between the two linear arrays, and is taken in line with the two linear arrays, as shown in Fig.(3.4).

The presence of a longitudinal gap needs to be accounted for when converting to a measured distance along the array. Given G as the longitudinal gap length, we update our calculations as follows:

$$\tau = \begin{cases} m\gamma, & m < 127 \\ m\gamma + G, & m > 127 \end{cases} \quad (3.9)$$

3. PEAK WIDTH CORRECTION

A longitudinal gap has effects on pairs of peaks that straddle the gap. We assume that the distance between two peaks is fixed, as shown in Eq(2.19). If a peak pair straddles the gap, we must add G to the distance between those peaks.

3.3.3 Transverse Gap

The transverse gap accounts for the distance between the two linear arrays that is perpendicular to the line of the arrays. The definition of the longitudinal gap is also shown in Fig.(3.4).

This has a similar effect on peak positions as the longitudinal gap. When converting to measured distances along the array, we account for the transverse gap as follows:

$$\tau = \begin{cases} m\gamma, & m < 127 \\ m\gamma + f(G, G_T) & m > 128 \end{cases} \quad (3.10)$$

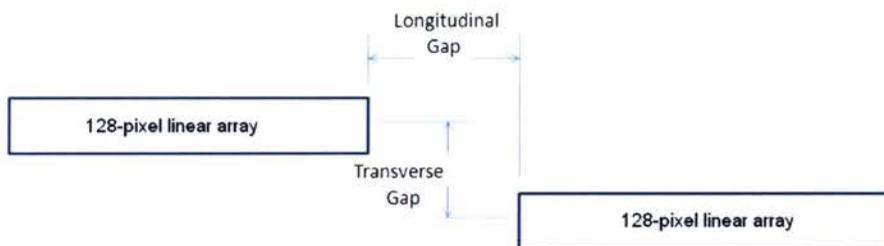


Figure 3.4: Definition of Longitudinal and Transverse Gaps.

3.4 Effects on Algorithm Performance

Because there is now some accounting for the changes in peak width across the array, it is expected that the results should improve where that quantity

is required, namely, in the NLSQ implementation. From before, the base image model is defined using the peak width in Eq(3.11), i.e., the equation

$$I_{mdl_0}(\tau) = e^{-\frac{\tau^2}{2\sigma^2}} \quad (3.11)$$

When a variable value of σ is used instead of a static value, it is expected that the overall error should decrease.

Given a static value of σ , the sun vector estimates are compared to the true sun vectors across the field of view. A plot of calibration error across the field-of-view is shown in Fig.(3.5). Here, the error grows as the sun-vector moves towards the outer edge of the field-of-view. Of particular note is a region of extremely high error on the upper right side of the field of view.

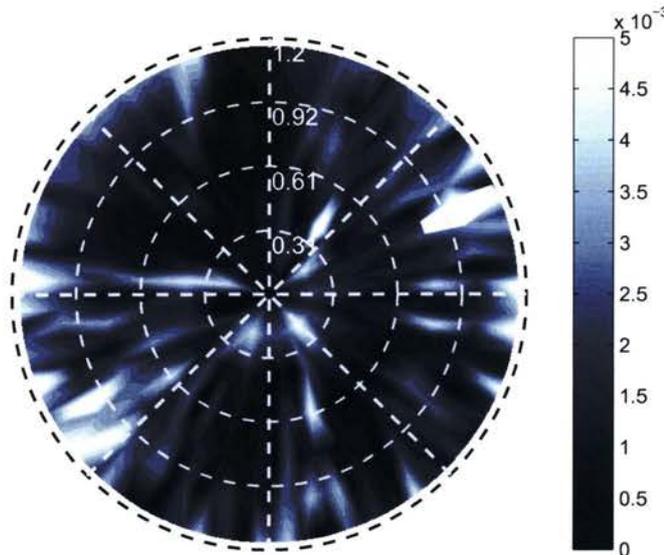


Figure 3.5: FOV Error Plot with No Width Correction (in Radians).

Conversely, a FOV error plot after width correction is shown in Fig.(3.6). When peak width correction is used, the error on the upper-right side of the field-of-view is much better resolved. There is also less differentiation in the

3. PEAK WIDTH CORRECTION

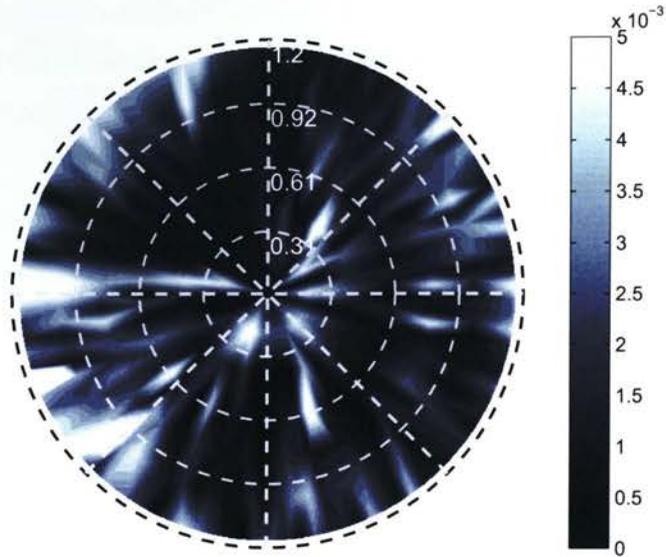


Figure 3.6: FOV Error Plot with Width Correction (in Radians).

error profile going from the central region towards the outside. Overall, the algorithm is more consistent in its ability to compute the sun-vector.

Integrating the changes that have been made to the calibration parameterization, we can see the evolution of the calibration error as the parameterization has changed. To illustrate the effects of the changes to calibration, a single calibration image set was used. This image set was used to calibrate an SS-411 sensor using progressively better calibration parameterizations. The effects of these changes to calibration are summarized in Table(3.2).

As the index of refraction and the gap parameterizations are added, there is a clear improvement in the mean error across the calibration set. The most notable improvement occurs when attention is given to the gap between the two parts of the linear array. Here, the error decreases by 35%. With the addition of the quartic width fit, the error drops by a small amount – 2.5% – compared to the same calibration routine using a constant value for σ .

Table 3.2: Calibration Summary

Params	Base	Refraction	Long. Gap	Trans. Gap	Width Fit
ρ_x	x	x	x	x	x
ρ_y	x	x	x	x	x
ϕ	x	x	x	x	x
ψ	x	x	x	x	x
h	x	x	x	x	x
\vec{r}_1	x	x	x	x	x
\vec{r}_2	x	x	x	x	x
tip/tilt	x	x	x	x	x
n_{glass}	1.51	x	x	x	x
G	—	—	x	x	x
G_T	—	—	—	x	x
σ	1.15	1.15	1.15	1.15	Quartic Fit
Mean Err. (mrad)	20.22	20.21	1.304	1.293	1.260

3.5 Summary

A method of relating the width of image peaks to the position of those peaks along the array was given. When this data was plotted, a trend in the data was seen. A fourth-order function was fitted to this data using least-squares error minimization techniques. Using this function with the NLSQ algorithm gave a slightly more consistent error performance.

The mean error of the NLSQ calibration using width data was compared to the same calibration with a constant width. There was a drop in mean error in radians of 2.5% when the width correction was included. This indicates that using the width correction step can lead to improved sensor performance.

3. PEAK WIDTH CORRECTION

Chapter 4

Precision Testing

Given a single sun-vector estimate, the error can be described as coming from two different sources. One source is an image-to-image error, due to limits in the sensor optics. We expect the image-to-image error to be a zero-mean random error. The other is error source is a bias term. Much of this term is due to sensor modelling inaccuracies. The purpose of the precision test is twofold. First, we would like to define the minimum angular motion that must take place before any motion can be recognized. The second goal of this section is to define some metric that describes how well the sensor registers relative motion between two points. When these points are close, much of the slowly-varying spatial bias should be eliminated.

Precision test results will be based on the repeatability of the sensor images. This means precision test data is specific to individual sensors - susceptibility to outside noise and dark-current noise vary from sensor to sensor, and these factors have a role in the precision test outcome. One error contributor that is particular to the test setup is errors in the motion of the test platform. Because the positioning of the test platform at a given point is also subject to error, this will have an effect on the precision data as well.

A number of steps are involved in performing the precision test. The first step in performing the precision test will be to compare a set of defined true

4. PRECISION TESTING

sun vectors to a set of estimates in order to find a bias error around a point of interest. By shifting the estimates by this amount, much of the slowly-varying spatial bias can be separated out of the error term. This will ensure that, for the remainder of the precision test, the estimation error will be due to sensor noise, algorithm limitations, and high-frequency bias variations. Further, it better allows for the assumption that the expected estimate is equivalent to the true value for sun-vectors where this correction has taken place.

It is expected that a number of images, when taken at a single point, will have some differences due to a number of noise sources. When these images are processed and translated into a sun vector, these estimates will differ by some small amount. By accounting for the bias error, the amount of estimation error caused by the random noise alone can be isolated. This will define the limits at which the sensor can resolve motion – and yields a method to find this limit.

The precision test also gives a sense of how well the the sensor can track small motions. Both the overall mean error and the relative error are of interest here. While the mean error will give a sense of how well the sensor performs in the absolute sense, the relative error tells how well the sensor tracks the true motion of the sensor, i.e., the smaller the relative error, the better the actual motion of the sensor is tracked.

4.1 Motion Limit

One thing that can limit the ability of the sun sensor to differentiate between two near points is the sensitivity of the detector to motion. Given the same target point, the sensor can return multiple slightly different images. This is due to small limits in the rotation platform, which exists only in the test situation, and slight varying ambient light conditions plus dark-current noise, which can exist in practice. Thus, even with perfect algorithms, a range

of sun vectors would be returned due to uncertainty in the measurement. Detectability of motion implies that a returned estimate offset is due to actual sensor motion, versus an offset caused by internal sensor noise.

4.1.1 Motion Limit Theory

In order to perform the motion limit test, a number of data sets must be gathered so that comparison can be done between the zero-motion set – a set of estimates taken with the same zero-shift true sun vector for all – versus a set where the sun vector has been shifted by some amount. The comparison is done using a 1-sided mean comparison test.

To do this, we consider offset angle θ as the random variable. This angle between a true sun-vector and it's corresponding estimate is found via:

$$\theta = \arccos \left(\frac{\hat{S}_{true} \cdot \hat{S}_{est}}{\|\hat{S}_{true}\| \|\hat{S}_{est}\|} \right) \quad (4.1)$$

The mean of the central-point set is forced to be an offset angle of zero. This allows for the computation of the following parameters for the two sets:

$$\bar{\theta}_0 = 0; \sigma_0^2 = \frac{\sum (\theta_{0,i} - \bar{\theta}_0)^2}{N_0 - 1} \quad (4.2)$$

$$\bar{\theta}_k = \frac{\sum \theta_{k,i}}{N_k}; \sigma_k^2 = \frac{\sum (\theta_{k,i} - \bar{\theta}_k)^2}{N_k - 1} \quad (4.3)$$

The null hypothesis will be that the two mean are equal, namely:

$$H_0 : \mu_0 = \mu_k \quad (4.4)$$

In order to perform the hypothesis test - which in this case, requires the student-t distribution - the degrees of freedom of the system needs to be found. This can be done using the following relation:

$$\nu = \frac{\left(\frac{\sigma_0^2}{N_0} + \frac{\sigma_k^2}{N_k}\right)^2}{\frac{(\sigma_0^2/N_0)^2}{N_0-1} + \frac{(\sigma_k^2/N_k)^2}{N_k-1}} - 2 \quad (4.5)$$

The data-weighted t value can be found using the means and variances pertaining to the two data sets by performing the following calculation:

$$\bar{t} = \frac{\bar{\theta}_k - \bar{\theta}_0}{\sqrt{\frac{\sigma_0^2}{N_0} + \frac{\sigma_k^2}{N_k}}} \quad (4.6)$$

Then, a confidence level is found, $\bar{\alpha}$, based on the data-weighted t value and the system degrees of freedom. This is done via the following relation:

$$\bar{\alpha} = F(\bar{t}|\nu) = \int_{-\infty}^{\bar{t}} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\sqrt{\nu\pi}} \frac{1}{\left(1 + \frac{t^2}{\nu}\right)^{\frac{\nu+1}{2}}} dt \quad (4.7)$$

Based on a confidence level at which to reject the null hypothesis, α , a comparison between the corresponding critical t value and the data-weighted t value from Eq(4.6) can be made. To find the critical value, the following calculation is performed - essentially the inverse of Eq(4.7):

$$t_{crit} = F^{-1}(\alpha|\nu) \quad (4.8)$$

By comparing the two values of t , the null hypothesis can either be confirmed or rejected:

$$\begin{aligned} \text{Accept } H_0 &\iff \bar{t} \leq t_{crit} \\ \text{Reject } H_0 &\iff \bar{t} > t_{crit} \end{aligned} \quad (4.9)$$

If the null hypothesis is rejected, then changes in the estimates can be attributed to actual motion of the sensor. If accepted, then the estimate offsets are considered to be caused by noise and image artefacts - not due to actual sensor motion.

4.1.2 Motion Limit Results

Testing for minimum detectable motion requires that multiple data sets be gathered. The first of these is of a number of images taken at the same point. These will return a number of sun-vector estimates centered around a point. The results from this set will be the reference against which all of the others will be compared. Other data sets will be taken in annular rings surrounding the reference point at a small distance. Each of these data sets will be compared to the reference set to determine whether or not the motion can be recognized, statistically, versus variance due to repeatability error.

For each annulus, a confidence level that motion has occurred was calculated. This value indicates the probability that detected motion is due to actual sensor motion instead of random measurement error. We note that the confidence level increases rapidly, approaching 100% at an offset angle of 4×10^{-3} radians.

The motion limit test was run for the data set shown in Fig.(4.3). For the purposes of this test, we use a confidence level of 95%. The values of $\bar{\alpha}$ found for the data sets in increasing order of angular offset are shown in Fig.(4.1). From the original data set, the first angular offset at which one can be confident that motion has occurred is at 2.62×10^{-4} radians. For the sensor, this means that when successive estimates are offset by less than this amount, one cannot be sure whether this is a result of actual sensor motion or simply a result of sensor noise.

4.2 Calibration Offsets and the Best Rotation Quaternion

Before precision testing can be performed, attention must be given to errors due to calibration. In order to perform accurate precision testing, the error must be due to repeatability error alone, and the estimates should not have

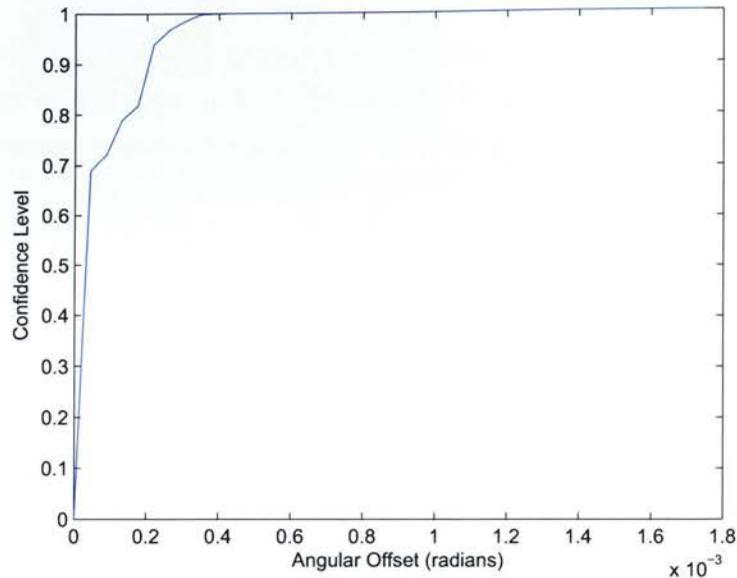


Figure 4.1: Confidence that Motion has Occurred.

a bias associated with the error term. As calibration error introduces a bias term, a method needs to be developed to remove these errors. An example of the difference between the true sun vectors and the estimates is given in Fig.(4.2). From the figure, we notice that the estimate vectors both have a random noise component as well as an overall bias error relative to the true sun-vectors.

In order to eliminate the bias in the small test area, a rotation quaternion will be found that maps the estimate field of points to the true sun vectors.

4.2.1 Quaternion Mapping Theory

The best rotation quaternion is used to relate a group of estimates (i.e., those given by the NLSQ or linear-phase algorithms based on actual sensor data) to those that yield a minimum total angular error when compared to the theoretical results. This is used to remove overall bias error in the data.

4.2. Calibration Offsets and the Best Rotation Quaternion

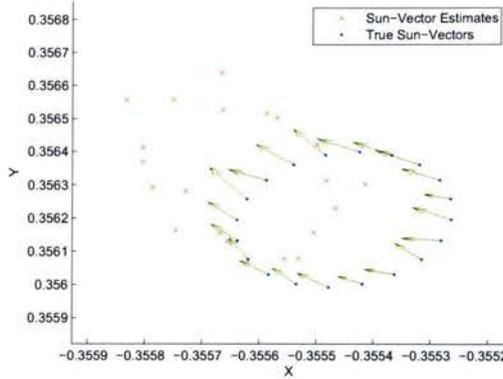


Figure 4.2: Difference Between Estimated and True Sun Vectors Before Bias Reduction.

The method used to find the desired quaternion is that suggested by Besl and McKay[Besl and McKay, 1992], which finds a quaternion that maps an altered set of data points onto a second set such that the error is minimized in the least-squares sense.

The quaternion is defined as a four-element vector $\vec{q} = [q_\theta \ q_x \ q_y \ q_z]^T$ subject to the constraints $q_\theta \geq 0$ and $q_\theta^2 + q_x^2 + q_y^2 + q_z^2 = 1$ [Hughes, 2004][Horn, 1987]. The two sets of data points under consideration are the set of N measured sun-vectors $P = \{\vec{p}_i\} = \{\hat{s}_{est}\}$, also known as the *estimates*, and the set of corresponding true sun vectors $X = \{\vec{x}_i\} = \{\hat{s}_t\}$. Given some rotation matrix R defined from the above quaternion, a scalar error (or cost) function can be defined by relating the two data sets[Besl and McKay, 1992]:

$$f(\vec{q}) = \frac{1}{N} \sum \|\vec{x}_i - R\vec{p}_i\| \quad (4.10)$$

The centroids of the data sets P and X can be found using a first-moment averaging of the vector elements across the set - namely, through the two relations

$$\vec{\mu}_P = \frac{1}{N} \sum p_i \quad (4.11)$$

$$\vec{\mu}_X = \frac{1}{N} \sum x_i \quad (4.12)$$

The cross-covariance matrix Σ_{PX} can then be defined that relates the estimated and true sun vectors.

$$\Sigma_{PX} = \frac{1}{N} \sum [\vec{p}_i \vec{x}_i^T] - \vec{\mu}_P \vec{\mu}_X^T \quad (4.13)$$

The cross-covariance matrix defined in Eq(4.13) allows us to form a corresponding 3x3 anti-symmetric matrix A via:

$$A = \Sigma_{PX} - \Sigma_{PX}^T \quad (4.14)$$

By extracting three elements from the anti-symmetric matrix A , a column vector $\Delta = [A_{23} \ A_{31} \ A_{12}]^T$ can be formed. This column vector, in conjunction with the skew-symmetric matrix, Σ_{PX} , allows us to define a 4x4 matrix Q (Σ_{PX})[Besl and McKay, 1992]:

$$Q = \begin{bmatrix} tr(\Sigma_{PX}) & \Delta^T \\ \Delta & \Sigma_{PX} - \Sigma_{PX}^T - tr(\Sigma_{PX}) I_{3 \times 3} \end{bmatrix} \quad (4.15)$$

The desired rotation quaternion is the eigenvector of the matrix Eq(4.15) which corresponds to the maximum eigenvalue.

4.2.2 Applying the Quaternion

The purpose in finding the best-rotation quaternion is to eliminate any large-scale bias errors in the data set. The method shown here is designed to relate two data sets in such a way that the magnitude of the error between them is minimized[Besl and McKay, 1992].

The quaternion itself is decomposed into angular and vector parts as follows:

$$q = q_\theta + q_x \hat{i} + q_y \hat{j} + q_z \hat{k} = q_\theta + \vec{q}$$

Related to this are two useful quantities, the vector \vec{a} and the angle ϕ , which can be derived directly from the quaternion by using the following relations:

$$\phi = 2 \arccos(q_\theta) \quad (4.16)$$

$$\vec{a} = \frac{\vec{q}}{\sin(\phi/2)} \quad (4.17)$$

These two quantities, once known, can be used to construct a rotation matrix that will correct the sun-vector estimates for overall bias. This rotation matrix can be calculated via the following relation:

$$R = \cos(\phi)I + \cos(\phi)\vec{a}\vec{a}^T + \sin(\phi)\vec{a}^\times \quad (4.18)$$

In the above equation, \vec{a}^\times is the skew-symmetric matrix created using the elements in \vec{a} . This rotation matrix is then used to modify the individual estimates in the annulus, using the multiplication

$$\vec{s}_{est,corr} = R\vec{s}_{est} \quad (4.19)$$

These annuli were then individually corrected by using a best-rotation quaternion. The true sun-vectors in each annulus are subjected to a constant offset ρ . An example of a pre-corrected set of data points is shown in Fig.(4.2) and the corresponding corrected estimates in Fig.(4.3).

When this correction is applied on the data set shown in Fig.(4.2), much of the bias that occurs in the area is removed. A corrected version of the data set is shown in Fig.(4.3).

Because the calibration bias is no longer an issue, this changes the ways in which the two data sets can be treated. Now, the corrected estimate

4. PRECISION TESTING

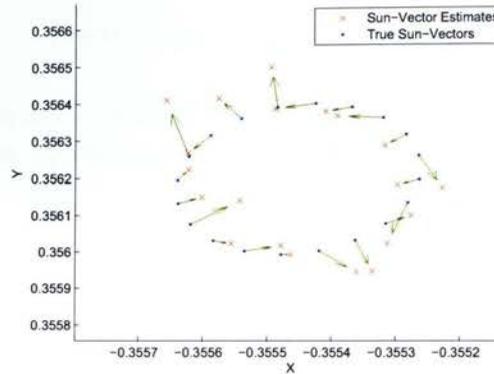


Figure 4.3: Difference Between Estimated and True Sun Vectors After Bias Removal.

offsets can be treated as random variables with a mean of *zero*, such that the zero-noise result would in fact be equal to the true sun-vector.

4.3 Precision

Once it has been established that true motion has been recognized, it would be useful to define how *well* the motion itself is detected. Given that motion has occurred between two points, the precision test is designed to give a comparison of the error in motion detection relative to the *actual* motion of the sensor.

The motivation for performing this test is to see how the *causes* of measurement error relate when the sensor is subject to small motions. Calibration error results indicate that the overall error does change gradually across the field of view, but is relatively constant when the scope is limited to a small area of interest. A typical calibration error level plot is shown in Fig.(4.4). As the region of interest becomes larger, the error due to calibration bias increases.

It is also important to note that relative error regarding the noise-induced

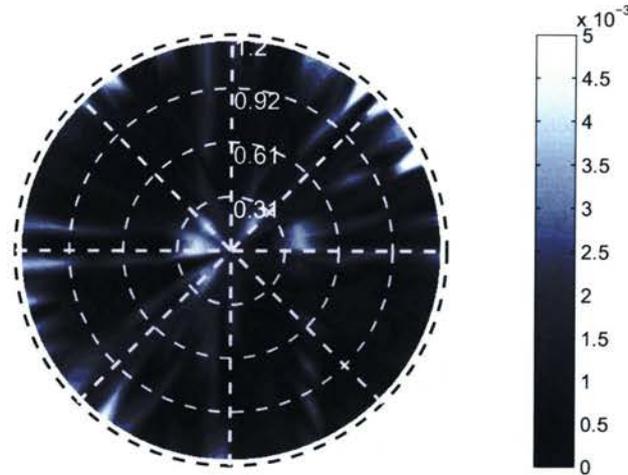


Figure 4.4: Typical FOV Error Plot (in Radians).

error relative to the actual sensor motion may be quite high when the angle of motion is small, due simply to the relatively small motion. The purpose of the precision test is to find how these two effects relate.

4.3.1 Definitions in the Precision Test

There are many different metrics that can be defined for the purpose of measuring precision. Here, variants of angular error calculations were used for comparing data sets. One of the first tasks required is to find the angular error between the corrected sun vector estimates and the true sun vectors, which can be found via the following relation:

$$\theta_{err} = \text{acos} \left(\frac{\vec{s}_0^T \cdot \vec{s}_{est}}{\|\vec{s}_0\| \|\vec{s}_{est}\|} \right) \quad (4.20)$$

This yields a scalar representation of error for each point. This can be

4. PRECISION TESTING

used alongside the true sun vectors to get a sense of error magnitudes as well as the variation of those error magnitudes across a field of points.

Our study used two different metrics for the calculation of local precision. First, a root-mean squared (RMS) angular error can be calculated for each annulus, which will give a sense of the angular error as the central offset increases. Given a length- n data set of errors $X = \{x_1, x_2 \dots x_n\}$, the RMS value is found via

$$\varepsilon_{rms} = \sqrt{\frac{1}{n} \sum x_i^2} \quad (4.21)$$

For our purposes, the data set was a vector of angular errors. This yields a scalar measure of error for a given data group.

The second type of metric used was a relative measure of angular error per unit offset, namely, the angular error encountered divided by the actual motion of the sensor. This is expressed as:

$$\varepsilon_{rel} = \frac{\varepsilon_{rms}}{\zeta_{true}} \quad (4.22)$$

4.3.2 Precision Test Results

A number of data sets were created in annular rings surrounding a central point. The intervals between each of these annular rings was broken down into three sections. The first of these sections was a series of annular rings taken in 2.5×10^{-3} degree intervals from the centre to a maximum of 0.1 degrees. The second section of rings was taken at 0.1 degree intervals out to a maximum of 1-degree offset, and the third at 1-degree intervals out to a maximum of 10 degrees. Each of these annular rings contained 20 points spread equally around the ring.

Plotting the RMS angular error versus the angular offset yields Fig.(4.5) and Fig.(4.6). From these figures, it is noted that the RMS error increases as the angular offset goes to 0.2×10^{-3} radians, at which point it levels off.

At an offset of 1.2×10^{-3} radians, the error begins to rise again. Further, as the angular offset becomes extremely large, the RMS angular error stabilizes at approximately 0.9×10^{-3} radians, as shown in Fig.(4.6).

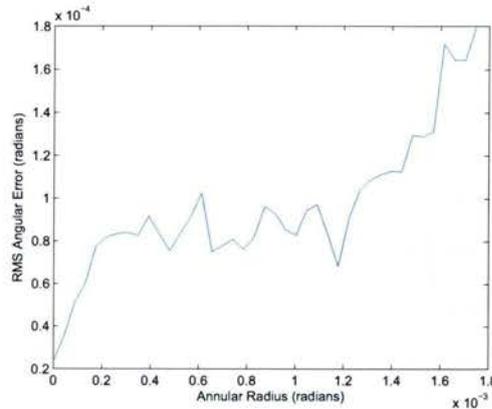


Figure 4.5: Local Region RMS Error.

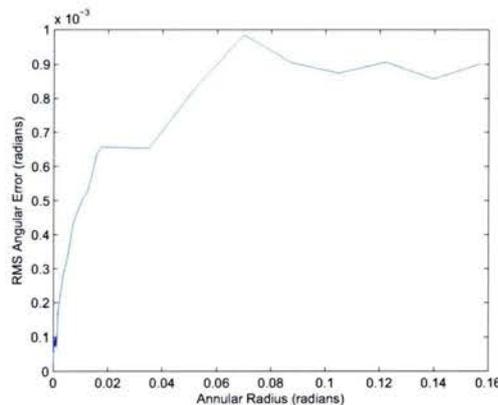


Figure 4.6: Far Field RMS Error.

It was expected that the RMS angular error would increase as the offset angle increased. This is because the spatial bias difference increases with the angle between the two vectors. At the intervals where absolute RMS error remains constant, however, the spatial bias error remains constant in those regions of angular offset.

4. PRECISION TESTING

Dividing the RMS error by the angular offset yields a relative measure of the estimation error. This relative measure is designed to get a sense of how precisely small sensor motions can be registered. From the RMS error plot in Fig.(4.5), it was noted that the RMS error remained relatively constant going from 0.2×10^{-3} radians to 1.2×10^{-3} radians, so it is expected that the normalized offset will be seen to decrease through that stage. The plot of the normalized error in Fig.(4.7) shows a steep drop that minimizes at approximately 1.2×10^{-3} radians. This is more pronounced when compared against the behaviour as the angular offset grows even larger, shown in Fig.(4.8).

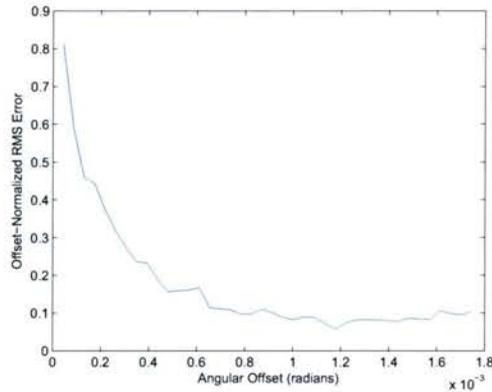


Figure 4.7: Local Region Normalized RMS Error.

As the offset becomes large, the offset-normalized error decreases simply due to this change. In the small-offset region, however, there is a distinct minimum that occurs. For this particular region of the field-of-view, this range corresponds to a region where the change in calibration bias is still very small, and the ability of the sensor to resolve small motions correctly is fairly high.

From the absolute RMS error metric and the normalized error metric, we can draw some important conclusions regarding the error behaviour. Where the normalized error is small, the sensor is good at resolving the sensor motion. This is because the spatially varying bias changes very slowly.

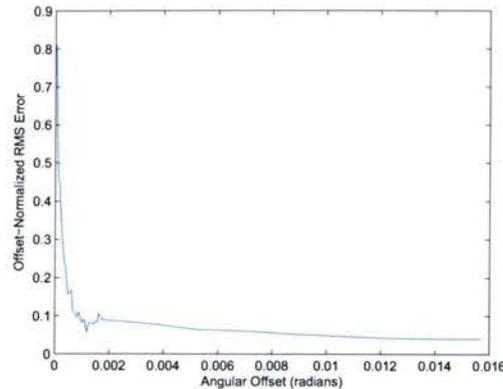


Figure 4.8: Normalized RMS Error to 1° Offset.

4.4 Summary

The purpose of the precision test was to provide a method of testing for sensor behaviour when subject to small motions. An algorithm was provided that eliminates calibration bias, which allows for the treatment of error as a zero-mean noise term once the bias correction has been made.

After bias correction, two tests were possible. First, a motion limit test was performed which defines the small motion limit at which the sensor is still able to register motion. Second, a number of metrics were developed that can be used to quantify precision at a particular point in the field-of-view. These two methods were demonstrated at a point and results of those tests were shown.

The motion limit test indicated that the smallest angular motion that can be observed by the sensor is 2.62×10^{-4} radians. If the sensor observes a motion less than this amount, we do not know whether the change is due to actual motion or to internal measurement error.

The precision test showed that the offset-normalized error minimizes at 1.2×10^{-3} radians. This indicates that the measurement of relative motion is extremely good here. Even though smaller motion can be recognized,

4. PRECISION TESTING

as the motion limit test showed, the sensor measures motion poorly until approximately 1×10^3 radians. As the offset angle grows large, the ability of the sensor to measure relative motion continues to improve.

Chapter 5

Filtering for Rate Sensing

In chapter 4, testing and calibration proved that results in a small, defined area had an overall bias, namely, the estimated sun vectors were skewed from the true sun vectors by a near-constant vector – so long as the field-of-view was limited to a small area. The purpose of this section is to explore how this recognition can be used to advantage, i.e., how one can account for this local bias to give better attitude estimation results. By using rate measurements, we hope to improve overall sensor performance.

Here, the types of observation are broken up into two parts. First, the classical single sun-vector measurement will be used. This, along with its associated error, will yield some attitude estimates over a period of time using a tumbling satellite. The second type of observation will further include a *relative* measurement, with its associated relative error. When relative measurements are taken, much of the slow spatially-varying bias error is removed. In this chapter, we use this to attempt to improve our attitude measurements.

The attitude dynamics themselves are inherently nonlinear, so an extended kalman filter (EKF) will be used to combine the observations with the expected attitude states. First, we will define the filter terms and the methods used to propagate the filter. Then the system itself will be described

using these filter terms, and initializations will be discussed. Using different combinations of the two types of observations, the performance of the filter will be analyzed and conclusions will be drawn based on this performance.

We will perform these simulations using two classes of sensors. The first of these will be a generic sensor for which no bias data will be used. This sensor will have a higher random error, and no spatially-varying bias. The second type of sensor will be based on the Sinclair Interplanetary sensors used in our calibration work. Here, the random error will be less than that of the generic sensor, but spatially-varying bias will be present. We will test the effects of the two types of observations on both of these sensors, resulting in a total of four simulations.

5.1 Filter Formulation

The EKF is designed to integrate observations made with propagated attitude states. Each observation updates the current state estimate. Based on the current observation and the filter history, a new gain is calculated at each time step that relates the amount of shift toward the observed state.

The EKF is designed specifically to handle nonlinear systems. Many of the terms in the filter equations are implicitly a function of the current state. Furthermore, many of the terms inside the filter equations are functions of time and need to be reevaluated at each filter step. The filter itself can be separated into two different parts – a continuous propagation of the state and covariance estimates with time, and a correction for the state and covariance that occurs with each new observation.

First, a number of matrices must be defined that are components of the filter. The first step in doing this is to define the state vector \mathbf{x} and the system equation vector \mathbf{f} such that the following continuous time relation is true[Stengel, 1994]:

$$\dot{x}(t) = f[x(t), u(t), t] + \Delta f[x(t), w(t), t] \quad (5.1)$$

Here, \mathbf{u} defines the known, applied forces, and \mathbf{w} the unknown disturbances. Essentially, \mathbf{f} defines those aspects of the system propagation, and Δf those aspects that aren't known.

In addition, another vector \mathbf{h} needs to be defined, which is the expected observation at each observation time step based upon the current state, i.e., the following variable needs to be defined:

$$Z(t) = h[x(t), t] \quad (5.2)$$

where

$$h[x(t), t] = Z[f(t), \Delta f(t)] = Z[x(t), u(t), w(t), t] \iff \Delta f = 0 \quad (5.3)$$

Defining the system equations allows one to define other terms which will be important to the filter. There are four matrices that we calculate in real-time that depend on the values of \mathbf{f} and \mathbf{h} :

$$F(t) = F[x(t), u(t), t] = \frac{\partial f}{\partial x} \quad (5.4)$$

$$G(t) = G[x(t), u(t), t] = \frac{\partial f}{\partial u} \quad (5.5)$$

$$L(t) = L[x(t), u(t), t] = \frac{\partial f}{\partial w} \quad (5.6)$$

$$H(t) = H[x(t), t] = \frac{\partial h}{\partial x} \quad (5.7)$$

Having defined these terms, one can now derive the filter equations. To propagate the state estimate and the covariance estimate, the following equa-

tions are solved in real-time[Stengel, 1994]:

$$\hat{x}[t_k(-)] = \hat{x}[t_{k-1}(+)] + \int_{t_{k-1}}^{t_k} f\{\hat{x}[\tau(-)], u(\tau), \tau\}d\tau \quad (5.8)$$

$$P[t_k(-)] = P[t_{k-1}(+)] + \int_{t_{k-1}}^{t_k} [F(\tau)P(\tau) + P(\tau)F^T(\tau) + L(\tau)Q_C(\tau)L^T(\tau)]d\tau \quad (5.9)$$

Here, Q_C is a matrix that satisfies the following relation regarding the noise:

$$E[w(t)w^T(t)] = Q_C(t)\delta(t - \tau) \quad (5.10)$$

Once new observations are received, the system is able to update the state and covariance estimates to account for the new data.

5.2 Defining the System

The states that have been chosen to be observed using the EKF are the body rotation rates \vec{w} , as well as the sun vector \vec{s} . The state vector has been defined as follows:

$$X = \begin{bmatrix} w_x \\ w_y \\ w_z \\ s_x \\ s_y \\ s_z \end{bmatrix} \implies \dot{X} = \begin{bmatrix} \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \\ \dot{s}_x \\ \dot{s}_y \\ \dot{s}_z \end{bmatrix} \quad (5.11)$$

The system dynamics themselves will be described using Euler's equations for rigid-body motion. The following equations for the body rotations will

be used:

$$\dot{w}_x = \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) w_y w_z + \frac{N_x}{I_{xx}} \quad (5.12)$$

$$\dot{w}_y = \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) w_x w_z + \frac{N_y}{I_{yy}} \quad (5.13)$$

$$\dot{w}_z = \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) w_x w_y + \frac{N_z}{I_{zz}} \quad (5.14)$$

The equations above define the first three terms of the vector f . For the last three terms, simply use the approximation

$$\dot{s}_i \approx \frac{\Delta s_i}{\Delta t} \quad (5.15)$$

To represent this purely in terms of state variables, it is important to note another useful approximation:

$$\Delta s \approx \vec{w} \times \vec{s}_i \Delta t \quad (5.16)$$

Therefore, one can expand Eq(5.15) to a representation purely in terms of state variables.

$$\vec{\dot{s}} = \begin{bmatrix} \dot{s}_x \\ \dot{s}_y \\ \dot{s}_z \end{bmatrix} = \vec{w} \times \vec{s} = \begin{bmatrix} w_y s_z - w_z s_y \\ -w_x s_z + w_z s_x \\ w_x s_y - w_y s_x \end{bmatrix} \quad (5.17)$$

This defines the last three terms of the vector f , satisfying the relation

$$f = f[X(t), u(t)] = \begin{bmatrix} \vec{w} \\ \vec{\dot{s}} \end{bmatrix}$$

5.2.1 Derivation of the Expected Observation

For this system, two different types of observations will be considered. In typical sun sensor usage, only the single sun vector observation is considered. We will compare the single sun-vector measurement to a velocity type measurement $\Delta s_k \times \Delta s_{k-1}$. In order to do this, one needs to find a value for the expected observation, h , in terms of the state variables \vec{w} and \vec{s} .

First, start with the basic relation

$$Z = h = (\Delta \vec{s})_n \times (\Delta \vec{s})_{n-1} \quad (5.18)$$

We know from Eq(5.16) that one can express $\Delta \vec{s}$ in terms of state variables and time, so rewrite h as

$$h = [\vec{w} \times \vec{s}_n \Delta t] \times [\vec{w} \times \vec{s}_{n-1} \Delta t] \quad (5.19)$$

By expanding this, one can arrive a slightly more useful relation

$$h = (\Delta t)^2 \{ [(-\vec{s}_n \times \vec{w}) \cdot \vec{w}] (-\vec{s}_{n-1}) - [(-\vec{s}_n \times \vec{w}) \cdot (-\vec{s}_{n-1})] \vec{w} \} \quad (5.20)$$

Here, two things are noted. First, the terms $(-\vec{s}_n \times \vec{w})$ and \vec{w} are perpendicular by definition, thus

$$(-\vec{s}_n \times \vec{w}) \cdot \vec{w} = 0 \quad (5.21)$$

Further, h cannot be expressed in terms of the old observation \vec{s}_{n-1} . Relating this to the current observation can be done simply by using the relation

$$\Delta \vec{s}_n = (-\vec{w} \times \vec{s}_n \Delta t) = \vec{s}_n - \vec{s}_{n-1} \quad (5.22)$$

Taking into account these changes, Eq(5.20) becomes:

$$h = -(\Delta t)^2 \{(-\vec{s}_n \times \vec{w}) \cdot [(-\vec{w} \times \vec{s}_n \Delta t) - \vec{s}_n]\} \vec{w} \quad (5.23)$$

Again, note two things that allow us to further simplify this equation. The dot product can be simplified partially by recognizing that the vectors $(-\vec{s}_n \times \vec{w})$ and $-\vec{s}_n$ are perpendicular, so that

$$(-\vec{s}_n \times \vec{w}) \cdot (-\vec{s}_n) = 0 \quad (5.24)$$

Further, the vectors $(-\vec{s}_n \times \vec{w})$ and $(-\vec{w} \times \vec{s}_n)$ are parallel, and of the same magnitude. This allows for the reduction of the equation to the simple usable form

$$h = -(\Delta t)^3 |-\vec{w} \times \vec{s}_n|^2 \vec{w} \quad (5.25)$$

5.2.2 Deriving F and H

Now that the value of h has been found, the values of the other filter parameters can now be derived. For this system, we assume no applied forces. As a result, $u = 0$ and $G = \frac{\partial f}{\partial u} = 0$.

In order to derive the value of F , note the values of f and X as follows:

$$f = \begin{bmatrix} \left(\frac{I_{yy} - I_{zz}}{I_{xx}} \right) w_y w_z + \frac{N_x}{I_{xx}} \\ \left(\frac{I_{zz} - I_{xx}}{I_{yy}} \right) w_x w_z + \frac{N_y}{I_{yy}} \\ \left(\frac{I_{xx} - I_{yy}}{I_{zz}} \right) w_x w_y + \frac{N_z}{I_{zz}} \\ w_y s_z - w_z s_y \\ -w_x s_z + w_z s_x \\ w_x s_y - w_y s_x \end{bmatrix} \quad (5.26)$$

5. FILTERING FOR RATE SENSING

$$X = \begin{bmatrix} w_x \\ w_y \\ w_z \\ s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.27)$$

In order to derive the value of F , simply take the Jacobian of the f vector with respect to X , as follows:

$$F_{6 \times 6} = \frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial w_x} & \frac{\partial f}{\partial w_y} & \frac{\partial f}{\partial w_z} & \frac{\partial f}{\partial s_x} & \frac{\partial f}{\partial s_y} & \frac{\partial f}{\partial s_z} \end{bmatrix} \quad (5.28)$$

While the derivation of F is fairly simple, the derivation of the H matrix is slightly more complex. Here again, take the Jacobian of the h vector with respect to X as follows:

$$H_{3 \times 6} = \frac{\partial h}{\partial X} = \begin{bmatrix} \frac{\partial h}{\partial w_x} & \frac{\partial h}{\partial w_y} & \frac{\partial h}{\partial w_z} & \frac{\partial h}{\partial s_x} & \frac{\partial h}{\partial s_y} & \frac{\partial h}{\partial s_z} \end{bmatrix} \quad (5.29)$$

The individual column vectors $\frac{\partial h}{\partial x_i}$ can be found using the product rule on the vector h , yielding the following result:

$$\frac{\partial h}{\partial x_i} = -2(\Delta t)^3 |-\vec{w} \times \vec{s}_n| \frac{\partial |-\vec{w} \times \vec{s}_n|}{\partial x_i} \vec{w} - (\Delta t)^3 |-\vec{w} \times \vec{s}_n| \frac{\partial \vec{w}}{\partial x_i} \quad (5.30)$$

The solution to $\frac{\partial \vec{w}}{\partial x_i}$ is trivial, in that it is either a unit vector (when x_i is a component of \vec{w}), or a zero vector (when x_i is a component of \vec{s}). A formulaic approach can be made for the solution of $\frac{\partial |-\vec{w} \times \vec{s}_n|}{\partial x_i}$ by first breaking down the cross-product as follows:

$$\vec{C} = -\vec{w} \times \vec{s}_n = \begin{bmatrix} w_z s_y - w_y s_z \\ w_x s_z - w_z s_x \\ w_y s_x - w_x s_y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad (5.31)$$

The cross-product magnitude can then be written as:

$$|-\vec{w} \times \vec{s}_n| = |\vec{C}| = \sqrt{c_1^2 + c_2^2 + c_3^2} \quad (5.32)$$

Now the more difficult derivative can be written in the form

$$\frac{\partial |\vec{w} \times \vec{s}_n|}{\partial x_i} = \frac{\partial |\vec{C}|}{\partial x_i} = \frac{1}{|\vec{C}|} \left(c_1 \frac{\partial c_1}{\partial x_i} + c_2 \frac{\partial c_2}{\partial x_i} + c_3 \frac{\partial c_3}{\partial x_i} \right) \quad (5.33)$$

This allows the algorithm to deal with the simple derivatives $\frac{\partial c_i}{\partial x_i}$.

5.2.3 Sensor Noise Modelling

Observation simulation will be conducted using two different sensor models. The first of these will be a generic sensor, and the second sensor model will be based on the SS-411 sensor. These will differ in how the observation error is handled. For both sensor types, the observer estimate model will be expressed as

$$\vec{s}_{obs} = \vec{s}_{true} + \vec{\varepsilon}_{obs} \quad (5.34)$$

For the generic sensor, there will be no bias error, and a large value of σ . Here, the magnitude of the error vector will be a function of a normally distributed random variable:

$$\|\vec{\varepsilon}_{obs}\| = \sqrt{2(1 - \cos \theta_{err})}, \quad \theta_{err} \sim N(\mu = 0, \sigma^2) \quad (5.35)$$

The direction of $\vec{\varepsilon}_{obs}$ will be determined randomly, and will not follow any distribution.

For the observer based on the SS-411, the error vector will be handled using a combination of bias and random noise. The bias will be provided from calibration data, and a smaller value of σ will be used. The error $\vec{\varepsilon}_{obs}$ then becomes

$$\vec{\varepsilon}_{obs} = \vec{\varepsilon}_{bias} + \vec{\varepsilon}_{norm} \quad (5.36)$$

Again, the direction of the normally distributed random error will be determined randomly, and the magnitude of the vector will be generated using

$$\|\vec{\varepsilon}_{norm}\| = \sqrt{2(1 - \cos \theta_{err})}, \quad \theta_{err} \sim N(\mu = 0, \sigma^2) \quad (5.37)$$

5.2.4 Planned Measurement Simulation

For this simulation, we will be comparing the observation results from two modelled sensors. The first of these will be a generic sensor, for which the observation error will be a zero-mean normally-distributed noise. There will be no spatial bias in the observations.

The second sensor we will model will be the SS-411 series sensors. In this case, there will be a spatially-varying bias term which we will model using calibration results. There will also be a normally-distributed noise component, however the standard deviation of the distribution will be much less than that of the generic sensor.

For each of these sensors, we will run the simulation using two different observation types. First, the standard sun-vector observation will be used. In the second case, the filter observation described by Eq(5.18) will be used in addition to the standard sun-vector measurement. In this case, the filter will use an combination of the current sun-vector measurement plus two taken previously to form the Z vector.

5.2.5 Initialization Values

Initialization parameters for the EKF are part guesswork and part theoretically derived. Although the initialization parameters can have only minimal

effects on the final output of the filter, a reasoned approach should exist for choosing these parameters to ensure correct filter operation.

We would like to define a system with an exact solution. This allows us to ensure that the simulation of system dynamics works correctly. In order to simplify the system such that this is the case, moments of inertia are chosen for a precessing satellite:

$$I_x = I_y = I_1, I_z = I_2 \quad (5.38)$$

Here, the satellite is axisymmetric. The advantage to this is that the angular displacements and velocities about the x- and y-axis can be described using an exact solution. The z-axis angular velocity should remain constant.

For the system presented here, the I vector was given the following value:

$$I = \begin{bmatrix} 10.1 \\ 10.1 \\ 19.8 \end{bmatrix} kg \cdot m^2$$

For the initial rotational velocity, the following values were used:

$$w = \begin{bmatrix} 0.0471 \\ -0.0337 \\ 0.7105 \end{bmatrix} rad/s$$

For the initial velocity estimate, the true velocity plus some normally-distributed random initial error with a standard deviation of $0.001^\circ/s$ was used.

5.3 Exact Solution

The free motion of an axisymmetric system can be described using an exact solution. An axisymmetric system has the following properties:

$$I_t = I_1 = I_2; \quad I_a = I_3 \quad (5.39)$$

Given these definitions of axial and transverse moments of inertia, the motion of the system can be described via [Hughes, 2004]

$$\omega_{3_0} = \omega_3 = v = \text{constant} \quad (5.40)$$

We can then write the relations for the other two angular rates. First we define a term Φ as

$$\Phi = \left(\frac{I_t - I_a}{I_t} \right) v \quad (5.41)$$

The equations to describe the other angular rates is

$$\omega_1 = \omega_{1_0} \cos(\Phi t) + \omega_{2_0} \sin(\Phi t) \quad (5.42)$$

$$\omega_2 = \omega_{2_0} \cos(\Phi t) - \omega_{1_0} \sin(\Phi t) \quad (5.43)$$

5.4 Generic Sensor Simulation

The generic sun sensor is designed such that the observation error is normally-distributed, the mean value being the correct observation. The observation offset is calculated using the error in Eq(5.35). For the standard deviation of a single sun-vector measurement, a value of $\sigma = 0.07^\circ$ was used.

5.4.1 Sun-Vector Observation Only

In this simulated case, we use the single sun-vector measurement only. Here, the observation Z is simply

$$Z = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.44)$$

Only the standard sun-vector measurement was used in this case. Error vectors between the true sun-vector and the corresponding state estimate were generated. The magnitude plot of this vector over time is shown in Fig.(5.1). Here, a significant amount of time was required for the initialization errors to be removed. The mean error vector magnitude was calculated, taking only values after 60s to account for the time to settle. The mean magnitude result in this case was 1.2×10^{-3} .

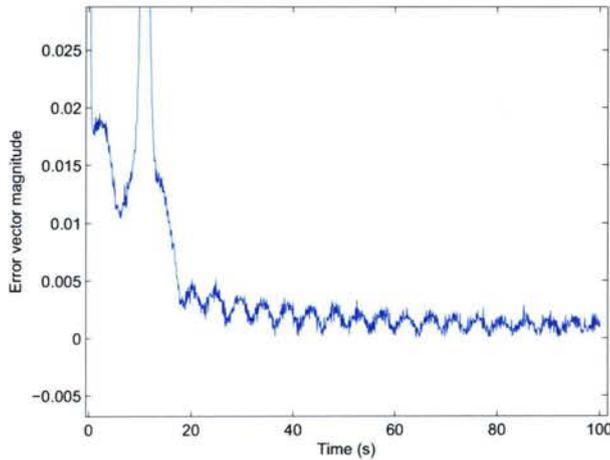


Figure 5.1: Error Vector Magnitude in \hat{s} for Generic Sensor, No Relative Observation.

5.4.2 Relative Observation

In this case, we use both the single sun-vector measurement as well as the relative observation expressed in Eq(5.18). In this case, the observation Z becomes

$$I_t = I_1 = I_2; \quad I_a = I_3 \quad (5.39)$$

Given these definitions of axial and transverse moments of inertia, the motion of the system can be described via [Hughes, 2004]

$$\omega_{3_0} = \omega_3 = v = \text{constant} \quad (5.40)$$

We can then write the relations for the other two angular rates. First we define a term Φ as

$$\Phi = \left(\frac{I_t - I_a}{I_t} \right) v \quad (5.41)$$

The equations to describe the other angular rates is

$$\omega_1 = \omega_{1_0} \cos(\Phi t) + \omega_{2_0} \sin(\Phi t) \quad (5.42)$$

$$\omega_2 = \omega_{2_0} \cos(\Phi t) - \omega_{1_0} \sin(\Phi t) \quad (5.43)$$

5.4 Generic Sensor Simulation

The generic sun sensor is designed such that the observation error is normally-distributed, the mean value being the correct observation. The observation offset is calculated using the error in Eq(5.35). For the standard deviation of a single sun-vector measurement, a value of $\sigma = 0.07^\circ$ was used.

5.4.1 Sun-Vector Observation Only

In this simulated case, we use the single sun-vector measurement only. Here, the observation Z is simply

$$Z = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.44)$$

Only the standard sun-vector measurement was used in this case. Error vectors between the true sun-vector and the corresponding state estimate were generated. The magnitude plot of this vector over time is shown in Fig.(5.1). Here, a significant amount of time was required for the initialization errors to be removed. The mean error vector magnitude was calculated, taking only values after 60s to account for the time to settle. The mean magnitude result in this case was 1.2×10^{-3} .

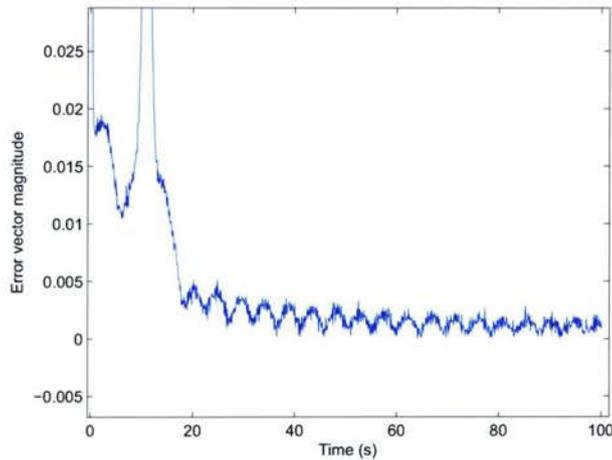


Figure 5.1: Error Vector Magnitude in \hat{s} for Generic Sensor, No Relative Observation.

5.4.2 Relative Observation

In this case, we use both the single sun-vector measurement as well as the relative observation expressed in Eq(5.18). In this case, the observation Z becomes

5. FILTERING FOR RATE SENSING

$$Z = \begin{bmatrix} [(\Delta \vec{s})_n \times (\Delta \vec{s})_{n-1}]_{3 \times 1} \\ s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.45)$$

Comparing one element of the true relative observation to the expected relative observation yields Fig.(5.2). Because the measurements in this case are extremely noisy, it is expected that the addition of the relative observation will do very little to improve the state estimates.

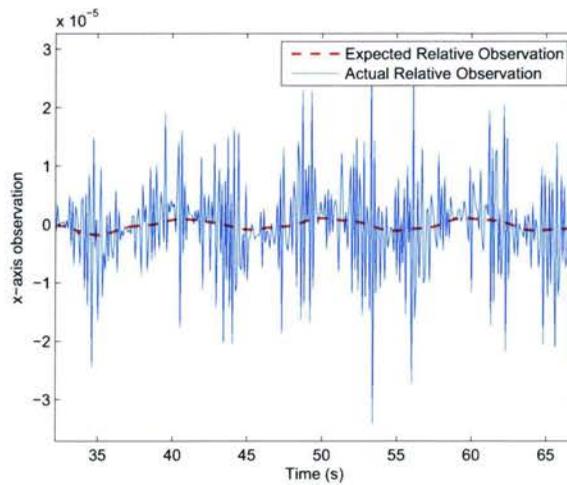


Figure 5.2: X-axis relative observation Comparison.

Creating the error magnitude term that relates the state estimates to the true sun-vector, and plotting with respect to time yields Fig.(5.3). This plot indicates that the introduction of the relative observation causes an increase in error. This is reflected in the mean error magnitude when taken after 60s - an increase in value to 3.2×10^{-3} .

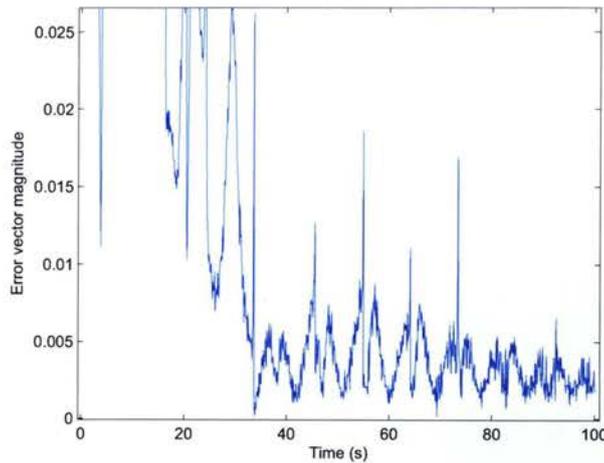


Figure 5.3: Error Vector Magnitude in \hat{s} for Generic Sensor with Relative Observation.

5.5 SS-411 Simulation

In these simulations, the SS-411 error term is modelled differently than that of the generic sensor. Here, the error is the sum of a bias term and a normally-distributed noise term. This offset is calculated using Eq(5.36) and Eq(5.37). Here, σ is the very small image-to-image noise result from the stationary sensor – 0.005° . The calibration bias will be taken from previous calibration work. An example of the slow calibration bias across the FOV was shown previously in Fig.(3.5) and Fig.(3.6). Note that, in practice, it is not generally possible to have this error map as it was generated based on exact knowledge of what the true sun-vector actually is.

5.5.1 Sun-Vector Observation Only

The single sun-vector measurement observation using the SS-411 model is the same as that for the generic sensor. The observation Z is again

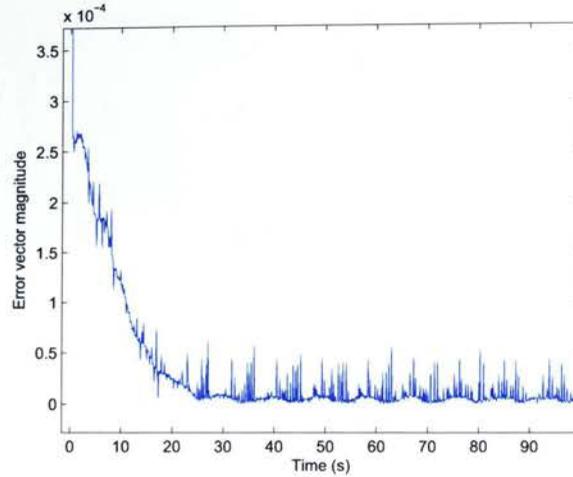


Figure 5.4: Error Vector Magnitude in \hat{s} for SS-411, No Relative Observation.

$$Z = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.46)$$

Generating the error vector magnitude through the use of the state estimate and the true sun vector yields Fig.(5.4).

The filter demonstrates that some time is needed to settle. Removing errors from the initialization, the mean error vector magnitude was calculated, ignoring the first 40 seconds of data to account for this settling time. The result here was a mean error magnitude of 7.99×10^{-6} .

5.5.2 Relative Observation

Again, the observation term Z is calculated the same way as was shown for the generic sensor with relative observation. The observation is found via:

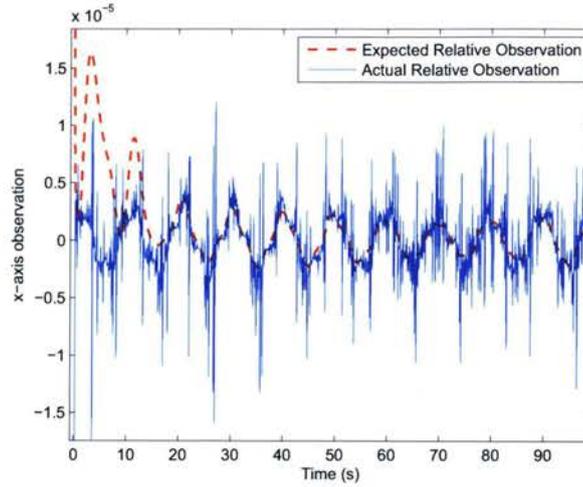


Figure 5.5: X-axis relative observation Comparison.

$$Z = \begin{bmatrix} [(\Delta \vec{s})_n \times (\Delta \vec{s})_{n-1}]_{3 \times 1} \\ s_x \\ s_y \\ s_z \end{bmatrix} \quad (5.47)$$

Comparing one of the relative elements of the expected observation versus the true observation yields Fig.(5.5). Here, noise in the relative observation is still extremely high; however, the actual observation can be seen to follow a similar shape to the expected observation.

Comparing the true sun-vector to that provided by the state estimate, an error magnitude term was created. Plotting the error magnitude term over time yields Fig.(5.6). Again, some time was required for the filter to remove initialization errors.. The mean error vector magnitude was again calculated for 40s and beyond, to account for this settling time. Here, the mean value was 7.70×10^{-6} .

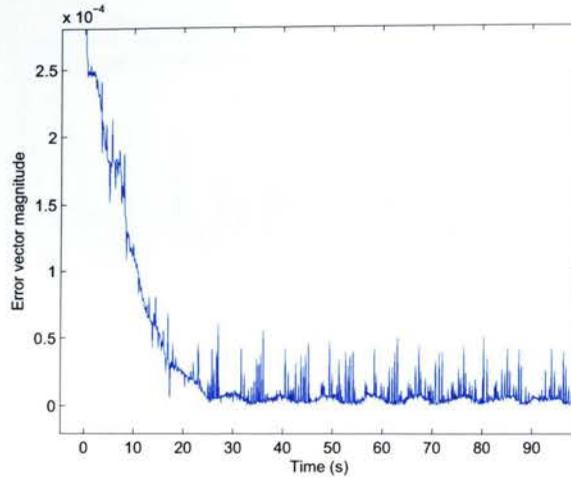


Figure 5.6: Error Vector Magnitude in \hat{s} for SS-411 with Relative Observation.

5.6 EKF Remarks

The results from the EKF tests indicate that the use of relative measurements has detrimental effects on the filter behaviour when the random observation error is high. Using the generic sensor model, the introduction of the derived relative observation worsened performance - increasing the steady-state mean error by 166%.

The EKF test using the SS-411 observer model showed a slight increase in performance; the steady-state mean error decreased by 3.6%. In this model, the random observation error component was small, but there was a significant spatially-varying bias contribution to error.

These simulations indicate that there is the potential to improve sensor performance by introducing a relative observation. For this to work, however, the random error component must be small. As long as the bias varies slowly across the FOV, the relative observation should ignore much of the contribution of this error component.

Chapter 6

Conclusions

The purpose of this thesis was to both better model the SS-256 and SS-411 sensor behaviour and understand sensor performance. Improving our sensor models improves sensor performance and yields better calibration results. Understanding sensor behaviour tells us what the limitations of the sensor are, and tells us how the sensor behaves in different conditions.

Included in this investigation was the derivation of a model describing the peak-width variation that occurs throughout the field-of-view. Also, the ability of the sensor to resolve motion was explored, including both small-angle detectability limits and a method of defining the precision of the sensor in detecting motion. Last, a new type of relative motion observation was defined and integrated into an extended kalman filter. We explored results for a generic sensor with normally-distributed noise only, as well as a SS-411 model where the error was treated as a combination of bias and random noise.

6.1 Summary

This section will reiterate the objectives of each section of the thesis, with some results obtained throughout the implementation of the developed meth-

ods.

6.1.1 Calibration Work

A physical description of the sensor used in the SAIL test facility, including a description of the laboratory setup, as well as a definition of the parameterization that was used for calibration. The calculations that are used to translate mean peak-positions in the image to a sun-vector observation were given.

A number of different super-resolution algorithms were suggested. The mathematics that convert raw image data to peak-positions were shown for each. In some cases, the coarse location of the peaks must be known so that the contributions from each slit-pair can be separated or the peaks can be coarsely bounded. A simple algorithm was shown that can perform this function.

An algorithm was also given that can generate image locations such that there is even coverage across the field of view. This is desirable so that calibration is not skewed in favour of particular sections in the field of view. Then, the calibration procedure was outlined, including the calibration parameters that are related to the mounting of the sensor, as well as those which are unique to each individual sensor.

6.1.2 Width Correction

The NLSQ algorithm requires that a Gaussian peak-shape model be matched to peaks in the array to determine the peak locations. Previously, a constant peak-width term was used to do this matching; however, data collected from peaks across the array in an image set show that this description is inadequate.

A full Gaussian model was derived that was used to find the peak-width as a function of array position. Given a defined error term, a least-squares

error minimization was used to define the width of the peaks. When the peak width was plotted against array position, a trend was observed when observations were limited to a single logical peak.

To describe this trend, a fourth-order even function was defined. Using the peak-width versus array position data obtained during the previous fit, the scaling terms of the function, as well as the minimum position, can be found through a least-squares error minimization. The curve fit for a particular logical peak map was shown to demonstrate this method.

Finally, the effects of the width-correction method on the calibration performance of the NLSQ algorithm was shown. Some very poor results observed near the edge of the field-of-view were shown, and the error profile was more consistent across the field-of-view.

6.1.3 Precision Testing

A method of testing how capable the sensor is at correctly identifying motion was developed. Although calibration results have been obtained before which describe a single estimation error obtained at points across the field-of-view, there was no previous method of finding the motion limits at which the sensor was usable, and it was unknown how well small sensor motions were tracked.

Because the calibration bias varied little when the area of interest was small, removing the bias error from a given point would limit the observed offsets to zero-mean noise. A best rotation quaternion that removes this bias error was derived. The application of this quaternion to the sets of estimates was demonstrated.

Because the sensor operation introduces random noise into the image, there will be a small-motion limit at which the sensor is usable, namely, there is a minimum angle at which the difference in two estimates can be attributed to actual sensor motion as opposed to random noise. A method of testing for this small limit was given, and an example of this procedure in use was provided. Using an example data set, a small motion limit of

6. CONCLUSIONS

2.62×10^{-4} radians was found.

The ability of the sensor to correctly identify motion was addressed in the precision test. Precision was defined using both an RMS angular error, as well as a normalized RMS angular error. Data sets were compiled using annular rings offset by small amounts from a central point. The RMS angular error generally increased as the offset angle became large; however, the error was relatively constant in the interval $[0.2 \text{ } 1.2] \times 10^{-3}$ radians and at offsets greater than 0.9×10^{-3} radians. The offset-normalized error dipped to a local minimum at approximately 1.2×10^{-3} radians and also dropped as the offset angle became large.

6.1.4 Extended Kalman Filter

There were two reasons an observation simulator was developed. First, a new type of observation was developed that looks at relative motion in addition to current position, and the extended kalman filter (EKF) would indicate whether or not this does improve sensor performance. Second, the simulator would indicate whether performance can be improved if the error profile of the sensor is well-known.

First, an EKF was defined that will propagate the state and covariance matrices continuously and will perform discrete updates to the state and covariance when observation data are available. We define a relative observation using the current sun-vector along with the previous two observed sun-vectors. For each of the two sensor types, we perform two observation simulations. In the first one, we use a single sun-vector measurement. In the second simulation, we use the single measurement as well as the relative measurement.

The simulator was first run using a generic sun sensor for which few error data were known as the observer. Using the relative observation in addition to the standard sun-vector observation gave worse results, increasing the steady-state mean error magnitude by 166%.

When the sensor error profile is well-known, however, better results can be generated from the filter. Taking advantage of the fact that the bias error is relatively constant when consecutive sun-vector movements are small, the filter is able to decrease the mean steady-state error magnitude term by a small amount, namely, 3.6%.

The results of this test indicate that relative measurements are only useful when random error is low. The generic sensor, with a large random error but no measurement bias, was unable to improve performance by providing relative measurement data. Simulation results improved with relative measurements provided by the SS-411 model. Although there was a large bias error in the model, the relative measurement removed most of this slow spatial bias. Since the random error of this sensor was small, the relative measurements provided were able to improve performance.

6.2 Future Work

Most of the work covered in the thesis focuses on the development of methods for improving knowledge of sensor parameters and sensor behaviour. Although examples of how to apply each of these methods are given, by no means are these results to be interpreted as representative of all Sinclair Interplanetary sensors. Indeed, the behaviour of each of these sensors is unique.

6.2.1 Field-of-View Precision

Although an example is given of the application of the precision test, the results do not hold even when different segments of the sensor field-of-view are examined. In order to get full understanding of the sensor precision, this test should be conducted at points throughout the field-of-view to get a more comprehensive look at the sensor behaviour.

6. CONCLUSIONS

Further, results from the precision test are algorithm dependent. Because much of the calibration bias is removed prior to the testing, the differences in estimation are primarily due to image noise. Depending on the ability of algorithms to reject this added noise, different results may occur.

6.2.2 EKF/Precision Model Unification

In order to achieve extremely high accuracy in modelling the sensor observations, one can merge results from a comprehensive precision test in the simulation, namely, performing the precision test throughout the field-of-view will yield expected error results from observation points around the field-of-view. Integrating these results with the simulator would give a more accurate simulation of the sensor behaviour and would yield a more accurate assessment pertaining to the ability of the sensor to track vehicle motion.

6.3 Closing Remarks

Sensor behaviour prediction was improved by introducing a peak-width function that can be utilized in conjunction with the NLSQ algorithm. This change is more consistent with the peak width behaviour across the field-of-view. A method for defining the sensor observation limits regarding small motions was given and demonstrated at a particular point. Metrics defining sensor precision were given, and the ability of the sensor to resolve small motions correctly was explored. Finally, an EKF was developed to simulate sensor observations and was tested utilizing both an arbitrary sensor for which only basic information is given, as well as a sensor for which intimate knowledge of the error profile is available.

Bibliography

- P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- Yam A. Li C. Enright, J. Modelling and testing of two-dimensional sun sensors. *IEEE Aerospace Conference*, 1:1–11, 2007.
- Yam A. Li C. Enright, J. Parametric processing for two-dimensional digital sun sensors: Algorithms, modeling and testing. *AIAA Journal of Spacecraft and Rockets*, 45(2):359–369, 2008.
- Zerubia J. Berthod M. Foroosh, H. Extension of phase-correlation to subpixel registration. *IEEE Trans. on Image Processing*, 11(3):188–200, 2002.
- Godard and J. Enright. Optimization of a sun-sensor illumination pattern using genetic algorithms. *IEEE Canadian Conference on Electrical and Computer Engineering*, 1:1437–1441, 2006.
- B. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, 4(4):629–642, 1987.
- Peter C. Hughes. *Spacecraft Attitude Dynamics*. Dover Publications, Mineols, New York, 2004.
- C.C. Liebe. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2), 2002.

6. CONCLUSIONS

- S. Mitra. *Digital Signal Processing: A Computer Based Approach*. McGraw-Hill, New York, New York., 3 edition, 2006.
- G. Rufino and D. Accardo. Enhancement of the centroiding algorithm for star tracker measure refinement. *Acta Astronautica*, 53(2), 2003.
- R. Stengel. *Optimal Control and Estimation*. Dover Publications, Mineola, New York, 1994.
- Larson W. Wertz, J. *Space Mission Analysis and Design*. Microcosm Press and Kluwer Academic Publishers, 3 edition, 1999.